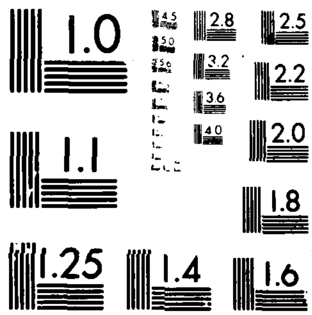
AD-A081 757 WESTERN UNION TELEGRAPH CO MCLEAN VA GOVERNMENT SYST--ETC F/G 17/2
COMPUTER PROGRAM DEVELOPMENT SPECIFICATION TERMINAL-TO-HOST PRO--ETC(U)
FEB 80 DCA200-C-637
UNCLASSIFIED SBIE-AD-E100 340 NL

1 OF 4
AD
A081757



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12) LEVEL II

AUTODIN II

See 1473 in back.

western union

Prepared Under Contract
DCA 200-C-637

By

WESTERN UNION TELEGRAPH COMPANY
GOVERNMENT SYSTEMS DIVISION

7916 Westpark Drive
McLean, Virginia 2210

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AUTODIN II

(12) LEVEL II

FINAL

COMPUTER PROGRAM DEVELOPMENT
SPECIFICATION
TERMINAL-TO-HOST PROTOCOL (THP)

CDRL ITEM NO. B006

DOCUMENT CONTROL # DIN II-F-623-13B06-1124

PREPARED FOR:

DEFENSE COMMUNICATIONS AGENCY
DCA CONTRACT 200-C-637

FEBRUARY 20, 1979

REVISION 1 - JULY 4, 1979
REVISION 2 - FEBRUARY 8, 1980

WESTERN UNION TELEGRAPH COMPANY
GOVERNMENT SYSTEMS DIVISION
7916 WESTPARK DRIVE
MCLEAN, VIRGINIA 22102

UNCLASSIFIED 450 7 MAR 80

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-E100 340

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER DCA 200-C-637-P006	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Computer Program Development Specification Terminal-To-Host Protocol (THP) . Revision 2.		5. TYPE OF REPORT & PERIOD COVERED 9 FINAL Rept. 6. ~~PERFORMING ORG. REPORT NUMBER~~
7. AUTHOR(s) Western Union Telegraph Company		8. CONTRACT OR GRANT NUMBER(s) 15 DCA 200-C-637
9. PERFORMING ORGANIZATION NAME AND ADDRESS Western Union Telegraph Company Government Systems Division 7916 Westpark Drive, McLean, Virginia 22102		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency ATTN: Code 450 Washington, D.C. 20305		11 12. ~~REPORT DATE~~ 8 Feb[redacted]80 13. ~~NUMBER OF PAGES~~ 296
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Defense Communications Agency ATTN: Code 450 Washington, D.C. 20305 12) 31		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited 18) SBIE

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release; distribution unlimited 19) AD-E100 340

18. SUPPLEMENTARY NOTES

Review relevance five years from submission date

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Telecommunications, Data Communications Protocol, Switching Systems, Telecommunications Systems, Data Communications System, AUTODIN II

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This ~~The~~ document provides the functional specification of the Terminal-To-Host Protocol (THP) as used in the AUTODIN II packet switched Network. THP provides the user (human or process) interface into the AUTODIN II protocol structure. Through a THP Command Language (TCL), a user requests and controls Transmission Control Protocol (TCP) services, and THP data packaging and Network Virtual Terminal (NVT) services. The NVT service will permit limited data exchange between normally incompatible terminals and/or host processes. This document also contains many examples of the THP design and implementation in the AUTODIN II MCCU.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 7 MAR 80 410 111

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

THP SPECIFICATION

REVISION 2, FEBRUARY 8, 1980

LIST OF CHANGED PAGES

Title	B-1a
12	B-15
27	B-27
32	D-2
33	D-5
35	D-6
39	D-7
50	D-8
73	D-9
A-1	E-1
B-1	F-1

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

LIST OF APPENDICES

LIST OF TABLES

LIST OF FIGURES

SECTION 1 - GENERAL

1.1 SUMMARY

This document provides guidelines with which an AUTODIN II-standard Terminal-to-Host Protocol (TMP) can be implemented. Included in this specification are:

- TMP Network Protocol (TMP-TMP)
- Network Virtual Terminal (NVT) Concept
- TMP-User Interactions
- Option Negotiation Mechanism

There are certain concepts which are fundamental to the understanding of this specification: (1) AUTODIN II, (2) TMP's role in AUTODIN II, and (3) NVT. AUTODIN II provides a means by which diverse and geographically separate hosts (computers) and terminals may communicate. Users (hosts and terminals) need have no knowledge of physical characteristics (word size, line speed, link protocol, etc.) or limitations of another user to communicate with the other user; network components compensate for these differences. These AUTODIN II components fall into one of three general categories:

1. backbone components - Switch Control Module (SCM); Supervisory SCM (SSCM); Standby Processor (SB)

2. access components - Channel Control Unit (CCU); Terminal Access Controller (TAC) (note that using this and related specifications a user may develop his own access component)

3. Network Control Center (NCC)

The CCU and TAC are the components which provide general users access to AUTODIN II. User provided components, also in the access component category, could be a subscriber host or host front-end which supports the AUTODIN II interface protocols. The AUTODIN II interface protocols are:

- SIP (Segment Interface Protocol)
- TCP (Transmission Control Program)
- TMP (Terminal-to-Host Protocol)

SIP is required by all users in order to access the Packet Switch (SCM). Users may employ a SIP to access the SCM and then develop any other levels of protocol (TCP and/or TMP or equivalent or user desired protocols) to communicate via AUTODIN II within their own common-protocol user community. However, to communicate with users connected to AUTODIN II

via a TAC or CCU, other users must support SIP, TCP, and THP protocols that are fully compatible with those of AUTODIN II.

THP's role in AUTODIN II is to provide part of the compensatory service to users. The "user" in the case of THP is the CCU process (in the host) or terminal user (connected to a TAC) which is being serviced over a particular host-CCU or terminal-TAC channel. THP, and other user-specific processes, such as Host Specific Interface (HSI) in a CCU or Terminal Handler (TH) in a TAC, know the specific characteristics of each user to be supported. THP, TH, and HSI use this knowledge to process data coming from and going to the user. For THP, user characteristics are contained in a user profile, one per user (see Appendix E).

The concept of NVT defines an AUTODIN II-network internal standard bidirectional, character-oriented device. THP is responsible for converting from user's to NVT format all data going from his user to the network, and for converting from NVT to user's format all data coming from the network to his user. NVT conversion includes simulating, as required, any carriage control functions, such as: linefeed, formfeed, or vertical/horizontal tab, ensuring line width is not exceeded, and converting to local values certain special characters: carriage return (CR), linefeed (LF), formfeed (FF), horizontal tab (HT), vertical tab (VT), bell (BEL), backspace (BS), null (NUL), erase character (EC), and erase line (EL). All information required to perform NVT conversion is maintained in the user profile. NVT mode is the default mode for any connection until the two THPs involved agree to process the data differently. This NVT/non-NVT agreement is made through the option negotiation mechanism (see Paragraph 5.1).

The reader should be familiar with these concepts. More background information is available in the referenced documents (see Paragraph 1.2).

It should be noted that this document contains many examples which emphasize the HCCU design/environment. These are merely examples of one implementation of the THP protocol. It is assumed that the reader is aware that many design decisions were based on the specific requirements for a CCU or TAC, and that the discussion of THP protocol itself consists of those portions of this document which relate to THP-to-THP communication. The user-THP, HSI-THP, and THP-TCP interfaces shown here are specific to the CCU/TAC THP implementations.

1.2 REFERENCES

The following documents provide additional information and detail on AUTODIN II, it's components and predecessors.

1. TELNET Protocol Specification, Network Information Center (NIC) 18639, latest issue (included in number 2, below)

2. ARPANET Protocol Handbook, NIC 7104, latest issue

3. Stanford Research Institute (SRI) Terminal-to-Host Protocol Specification, 15 July 1976

4. SRI Transmission Control Protocol Specification, 15 July 1976

5. Defense Communications Agency System Performance Specification (Type A) for AUTODIN II Phase I, latest issue

6. Western Union (WU) Proposal WU-301-1 AUTODIN II Phase I, April 1976, latest issue

7. WU AUTODIN II Design Plan, latest issue

8. AUTODIN II Design Plan Executive Summary, latest issue

9. AUTODIN II TCP Transportable Specification, latest issue

10. AUTODIN II SIP Transportable Specification, latest issue

SECTION 2 - CONNECTION PROCESSING

2.1 GENERAL THP PROCESSING

The mechanics of establishing, maintaining, and terminating a virtual connection are the responsibility of TCP (Reference 9), but it is THP that requests TCP's services on behalf of the local user (host process of CCU or terminal user on TAC). THP is responsible for providing service to the user, both in processing the user's data going toward the network and in processing network data going to the user. Toward this end TAC/CCU THPs maintain certain data structures, reflecting user characteristics and the current status of the virtual connection.

Also included in this section is a brief discussion of the event mechanism, the method of communication between the various programs in the CCU and TAC. In addition, THP (connection) states and THP output messages to the user are discussed.

2.1.1 Data Structures

Although data structures could be considered to be implementation dependent, the major THP data structures are discussed here to point out the basic information about each user required to perform THP processing. There are four major data structures used by a TAC/CCU THP:

1. Connection Data Structure (CDS) - there is one such structure for each user and it contains the other three major data structures

2. Connection Management Block (CMB) - contained in the CDS, this structure has information about the user's connection, such as, current state, buffer information, etc.

3. User Profile Block (UPB) - contained in the CDS, this structure has information about the user's characteristics, such as, local character values, tab settings, etc.

4. Network Virtual Terminal Table (NVTTAB) - contained in the CDS, this structure provides a character "lookup" for each 7-bit ASCII character received from the user; the value associated with each character directs THP's required action for that character value.

The MCCU data structures are shown in Appendix E. The SCCU and TAC data structures are not shown but are logically similar to those for the MCCU. That is, similar information is maintained for the user, although the structures themselves may be different.

2.1.2 THP Events

THP is driven by significant circumstances, such as, data coming in from the user, data coming in from the network, remote closing of the current connection, preemption of the current connection, and so forth. These significant situations are, for the most part, conveyed to THP by either TH, HSI, or TCP via inter/intra-process communications called "events" in the CCUs and TAC. Events are processed in turn and carry enough information to complete the task required for the significant situation. For example, the arrival of data from the user is conveyed in the MCCU by an HSI-to-THP From User event. The event consists of user and buffer information so that THP can process the data for the network. Events defined for the MCCU appear in Appendix F. These are conceptually similar to those used by the SCCU and TAC implementations, although the event format and mechanics of inter/intra-process communication may differ. Any reference to an event in this document refers to an MCCU event.

2.1.3 THP States

Based on user input or events from TH/HSI/TCP, THP changes the state of the user's virtual connection. There are eight states which have significance in the CCUs. THP "states" are defined here based on the MCCU implementation. It is conceivable that other implementations of THP may not require so many, or may require more states. In fact, the TAC THP maintains fewer states, a result of the overall TAC environment. The basic "virtual connection states" are as summarized below, however, whether realized or maintained by a specific THP implementation or not. These states are also shown in Appendix C:

1. disconnecting - a Line Disconnect event has been received from HSI when a connection was established, causing THP to send a request for an immediate close to TCP; when in this state THP is waiting for the close processing to complete

2. disconnected - hardware link (channel) associated with the user is not active; THP is waiting for

notification that the link has been initialized (Line Connect event)

3. aborting - the connection has been aborted for some reason (THP protocol error, user abort command, TCP direction); THP is waiting for abort processing to complete before going to closed state

4. closing - a close command was received from the user; THP is waiting for close processing to complete before going to closed state

5. active - open processing is complete and connection is completely established

6. opening - a request for a connection has been received from the user or an auto-open function has been initiated; THP is waiting for notification that the connection has been established (Open Complete event); data received from the user will be saved until the connection is established and then processed as normal

7. listening - a notice has been sent to TCP that the user is ready for a connection; THP is waiting for notification that a connection has been established (Open Complete event); data received from the user will be discarded and the user notified

8. closed - there is no established connection, nor has one been requested, but the hardware link associated with the user is still active

2.1.4 THP Output Messages

In order to inform the user of certain conditions concerning his connection or input data stream, THP will send an appropriate THP output message to the user. These messages are shown in Appendix C, including text as well as the reason for output to the user.

A user will be designated, in his profile, as receiving one of three message types. This indicator is changeable via the set command (see Paragraph 4.3.2). The three message types are:

1. verbose - the entire message text will be sent to the user, e.g., @<32>CONNECTION DOES NOT EXIST (where: "@" is prefix character defined for user)

2. concise - only the prefix character and message

number are sent to the user, e.g., <30>

3. none - no THP output messages will be sent to the user

2.2 ESTABLISHING A CONNECTION

An attempt to open a connection will be initiated by TMP on behalf of the user, (1) if the user enters a valid open command (see Paragraph 4.2.1) or (2) if the user enters data and is designated as an "auto-open" type (see Paragraph 4.2.3). TMP will send a request to TCP to establish a connection to a specific remote user. TCP is responsible for establishing, maintaining, and terminating virtual connections for AUTODIN II. Upon receipt of the open request (Open event), TCP will perform preliminary validation, but will not begin connection establishment until data is ready to be sent (a subsequent request (Send event) from TMP). Once TMP receives preliminary acknowledgement (Open Return event) from TCP for the open request, TMP open processing begins. It should be noted that a connection can be established under only two conditions. First, the specific remote user has a "listen" outstanding (see Paragraph 4.2.2), indicating he is willing to have the connection established. Second, the new connection preempts an existing connection between the specific remote user and a third subscriber, when appropriate conditions are met (see Paragraph 2.2.2).

TMP open processing begins with the creation of a characteristics option record (see Paragraphs 3.3 and 5.3.1). It is the sending (to TCP in a Send event) of this record that will cause the connection to be established (i.e., TCP's three-way handshake to take place). TMPs exchange characteristics during open processing for three reasons: (see also Paragraph 5.3.1)

1. to perform a user compatibility check according to the approved cross-connection matrix (see Table 1)

2. to determine if users have "identical user's characteristics" so that binary mode may be entered, providing efficient utilization of AUTODIN II resources (see Paragraphs 3.4, 3.5 and 5.3.2)

3. to determine if users are incompatible in the use of a go-ahead signal (see Paragraphs 4.4.6 and 5.3.5)

TABLE 1. CROSS-CONNECTION MATRIX

SEND \ RECEIVE	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Y	S	S	S	S	S	S	S	S	S	Y	Y	Y
2	S	X	X	N	X	X	X	X	N	N	S	S	S
3	S	X	X	N	X	X	X	X	N	N	S	S	S
4	S	N	N	X	N	N	N	N	X	X	S	S	S
5	S	X	X	N	X	X	X	X	N	N	N	S	R
6	S	X	X	N	X	X	X	X	N	N	N	S	R
7	S	X	X	N	X	X	X	X	N	N	N	S	R
8	S	X	X	N	X	X	X	X	N	N	N	S	N
9	S	N	N	X	N	N	X	X	N	N	N	S	R
10	S	N	N	X	N	N	X	X	N	N	N	S	R
11	Y	S	S	S	R	R	N	R	S	N	Y	Y	R
12	Y	S	S	S	R	R	N	R	N	N	Y	Y	R
13	Y	S	S	S	R	R	R	N	R	R	R	R	Y

LEGEND:

1. Non-CCU Host to SCM (SIP required)
2. CCU - TTY Type Port
3. CCU - Programmable CRT Type Port
4. CCU - Magtape/Card Type Port
5. TAC - CRT/TTY Terminal
6. TAC - Non-programmable Buffered CRT Terminal
7. TAC - Programmable (Mode I) Buffered CRT Terminal
8. TAC - Programmable (ADCCP) Buffered CRT Terminal
9. TAC - Magtape/Card (ADCCP) Terminal
10. TAC - Magtape/Card (Mode I) Terminal
11. Facsimile Terminal to SCM (SIP required)
12. Graphic/Light Pen (RAND) Terminal to SCM (SIP required)
13. Sensor Terminal to SCM (SIP required)

X - cross connection allowed
S - cross connection supported (assumes AUTODIN II THP/TCP supported by both user's access areas)
N - cross connection not supported
R - cross connection restricted (connection not allowed by AUTODIN II A-Spec (Reference 5)
Y - cross connection allowed; Host-to-Host protocol (TCP, THP, or other) compatibility must be determined by subscribers

Revision 1, July 4, 1979

Only the first check is required of THP. That is, the second and third checks are made only if a THP has implemented the characteristics option or is serving a user in need of go-ahead compatibility. Because characteristics checking is an option, the request to perform it may be rejected. The cross-connection matrix check, however, will be implemented by AUTODIN II standard THPs. This procedure is discussed in more detail in Paragraph 5.3.1. It should be noted that if the cross-connection matrix check fails, the connection will be closed immediately and both users will be notified.

Once characteristics exchange is complete, THP considers the connection to be active and processing ensues based on the outcome of characteristics option, i.e., binary or NVT mode (see Paragraphs 3.1, 3.4, 3.5 and 5.3.1).

2.3 TERMINATING A CONNECTION

A connection may be terminated by one of several actions, only one of which (number 1, below) guarantees the delivery of all data sent by the requesting user. The remaining actions cause connections to be aborted and the user must assume not all data entered was delivered to the destination user. The reasons for connection termination are:

1. close request from local/remote user (see Paragraph 4.2.4)

2. abort request from local/remote user (see Paragraph 4.2.5)

3. connection timeout (see Paragraph 2.3.1)

4. connection preemption (see Paragraph 2.3.2)

5. protocol error (see Paragraph 2.3.3)

6. cross-connection matrix violation (see Paragraphs 2.2 and 5.3.1)

2.3.1 Connection Timeout

Although timing certain conditions could be considered a choice of implementation, it is discussed here because AUTODIN II-standard TMPs and TCPs will include the procedure. Also, it seems appropriate to stress the importance of timeouts in providing more efficient service to the user. Conditions which will be monitored are summarized below. In all the below situations the connection will be closed immediately and each user will be notified by his TMP that the connection has been terminated.

1. inactivity on connection - This is detected by TMP when data has not been sent or received on the connection for a specified time. It also facilitates the situation in which a host will not be able to enter a close command. A recommended time duration is on the order of 5 minutes, but this will be established for each user during system generation and can vary. The user will be able to turn the timer off and on via the time command (see Paragraph 4.3.9) during an active connection. The connection inactivity timer will be started (if a value has been generated in the JPB) once the characteristics option negotiation has completed, and will continue until connection closure (unless turned off by the user).

2. inordinate time elapsed during option negotiation - This is detected by TMP. The assumption is made that the remote TMP has not responded for some reason and it is considered to be a possible protocol problem. The user has no control over this timeout value (approximately 3 minutes in duration). It will be started for all option negotiations whether the connection inactivity timer is running or not.

3. no TCP acknowledgement for data sent over network - This is detected by the local TCP when the remote TCP acknowledgement is not received for segments sent. After the timeout interval, TCP will retransmit the unacknowledged segment. This will be repeated three times. If an acknowledgement is still not received, the assumption is made that something is wrong with the remote user or the network itself. TCP will then abort the connection. The user has no control over this timeout value.

2.3.2 Connection Preemption

Connection preemption is handled by TCP, but the procedure is summarized here as TMP is indirectly involved. Each user (host process or terminal user) will be administratively designated as "preemptable" or "nonpreemptable." This designation will be in the TMP user profile and will not be changeable other than by system generation. The indicator will be made available to TCP and used to determine if preemption should be allowed. In addition, users connected to a TAC actively participate in the preemption process. The following set of circumstances defines a preemption situation: (all of these must exist for a preemption to occur)

1. segment received (by TCP) from network; must be "SYN" segment; one that would normally begin to establish connection as first leg of TCP three-way handshake

2. connection already established (not opening, listening, or closing) for local socket

3. foreign socket of segment received is different than for established connection

4. established connection is marked preemptable

5. new segment precedence is in a higher category (I,II,III,IV) than the precedence of established

connection

6. new segment security/TCC are valid for local socket

If the above set of circumstances exists, TCP will immediately close the established connection. Both users on the preempted connection will be informed that the connection was closed. The procedure for establishment of the "preempting" connection varies between the CCU and TAC implementations.

In the CCU, when the close processing is complete for the preempted connection, TCP will inform THP of the new connection via a Preempt event. THP will proceed with open processing for the new connection as if the Preempt event were notification of a connection establishment (Open Complete event). The user will be notified of the new connection when open processing is complete.

In the TAC, when the user receives the preemption notification, he must enter an open command sequence requesting the new connection, effectively, acknowledging the preemption. If a "matching" open request is not received within a certain time frame, the "preempting" connection is also closed. The "matching" information will be given in the preemption notice to the user.

2.3.3 Protocol Errors

Certain circumstances detected by THP will cause an existing connection to be closed immediately. These circumstances are considered to be violations of the THP-to-THP procedures and are, therefore, protocol errors, resulting in an abort of the existing connection. It is assumed that if these errors occur, the integrity of the connection is in jeopardy. There are currently eight THP protocol errors defined. All but one, number 8 below, relates to option negotiation; but all are situations in which one THP has detected an illogical sequence in the network data stream, i.e., one which is precluded by standard procedures. Because these procedures are clearly defined in this document, violations are considered catastrophic. There will be no procedure for acquiring user direction concerning continuance of the connection when these errors are detected. The following situations are treated as protocol errors when detected by THP:

1. receipt of an option negotiation record from the network requesting to start an option currently being performed, previously negotiated to start (see Paragraph 5.2)

2. receipt of an option negotiation record from the network requesting to stop an option that is not currently being performed, not previously negotiated to start (see Paragraph 5.2)

3. receipt of an option negotiation record which is a refusal to honor an option negotiation request to return to an NVT default condition, i.e., to stop a previously negotiated option (see Paragraph 5.2)

4. receipt of an RCTE control record from the network when RCTE option was not previously agreed to (see Paragraph 5.3.4)

5. receipt of an XASCII data record from the network when the XASCII mode option was not previously agreed to or when the connection is in binary mode (see Paragraph 5.3.3)

6. receipt of a goahead control record from the network when the goahead option was not previously agreed to (see Paragraph 5.3.5)

7. receipt of an NVT record (CR, LF, FF, HT, VT, BEL, BS) from the network when the connection is in binary scanning mode (this applies to record mode only; if in stream packaging mode and binary scanning mode, NVT values are passed as data)

8. receipt of data from the network which is not in record format when connection is in record mode (see Paragraph 3.5)

SECTION 3 - TMP DATA SCANNING/PACKAGING MODES

3.1 INTRODUCTION

The AUTODIN II standard terminal format is established in the definition of a network virtual terminal (NVT) model. This model represents an imaginary terminal but one of which the characteristics are known to all CCU, TAC, and fully conforming TMPs in the AUTODIN II network. With this knowledge TMPs are able to convert data for users, as required, going to and coming from the network. The NVT concept ensures that each TMP will know the format of data coming from the network, thus allowing each TMP to compensate for the special characteristics of his user. The procedures for scanning and formatting user data while in NVT mode, as well as the NVT model definition, are discussed in Paragraphs 3.2, 3.4, and 3.5. NVT is the normal or default processing mode for all connections. TMPs may agree to vary this processing, however, via option negotiation (see Paragraphs 5.1 thru 5.3). Two additional special modes which are related to the NVT concept (binary and XASCII), entered via option negotiation, are discussed in Paragraphs 3.4 and 3.5. An additional variance to NVT processing, transparent mode, which is a local (not negotiated with remote TMP) option, is also discussed in these paragraphs. The basic difference between the scanning modes is the subset of NVT-related characters which is recognized while in any specific scanning mode.

There are two methods of packaging user data going toward the network, record mode and stream mode. These modes are only indirectly concerned with the scanning modes, as will be explained in Paragraph 3.3.

Appendix G discusses the basic units of transmission for AUTODIN II, explaining the terminology used for each unit as well as the relationship of all the units. An understanding of these terms is important to the understanding of the subsequent paragraphs.

3.2 NVT MODEL

Data coming from the network will have been formatted by the source TMP as if destined for an NVT "printer." That is, of course, unless agreements to vary NVT default format have been made by the two TMPs via option negotiation (see Paragraph 5.3). There is no specified line width or page size for NVT printer. However, each TMP will perform certain functions to accomodate the line width and page size for the local user. The user's desired values are considered the NVT default values. Both line width and page size are subject to change via option negotiation. The NVT printer can produce representations of all 95 ASCII graphics (codes 32 thru 126). Of the 33 ASCII control codes (0 thru 31 and 127), the following have specified meaning to NVT printer:

1. carriage return (CR) - moves print head to left margin of current line

2. linefeed (LF) - moves print head to next printline keeping same vertical position

3. formfeed (FF) - moves print head to top of next page (leftmost column of first printline of page)

4. horizontal tab (HT) - moves print head to next horizontal tab stop; if print head is currently beyond last tab stop on line, HT moves print head to first column of next line, i.e., performs local new line function

5. vertical tab (VT) - moves print head to next vertical tab stop; leftmost column on that line; if print head is currently beyond last tab stop on page, VT moves print head to top of next page, i.e., performs local FF function

6. backspace (BS) - moves print head one character position towards left margin

7. bell (BEL) - produces an audible or visible signal which does not move print head

8. null (NUL) - provides timing delays but causes no function to be performed at printer

Each of these characters will be converted to an NVT record (for record mode) or an NVT model value (for stream mode) to be sent across the network. The NVT model value is the same as the NVT record type value (see Appendix A). The NVT default for each character will be the NVT record/value with no timing delays (nulls) following. It should be noted that the CR and LF records/values when in sequence represent the

NVT new line function. When these are received in sequence in the network data stream, they should be converted, as required, to the local new line sequence.

Other codes which do not effect the NVT printer, as such, but are considered part of the NVT model are data transmission control characters. These have meaning either upon receipt from the user or upon receipt from the network. A detailed explanation is found in Paragraph 4.4. These special characters are also listed and categorized here as each relates to the NVT model.

1. Send Now (SN) - causes accumulated user data to be sent to network (see Paragraph 3.6); the SN character is not sent across network (see Paragraph 4.4.2)

2. Are-You-There? (AYT) - causes a character or hardware signal, if defined, to be sent to remote user which presumably causes status from that user to be returned to initiator of AYT character; treated same as NVT characters, i.e., converted to/from NVT model value (see Paragraph 4.4.3)

3. Erase Line (EL) or Erase Character (EC) - causes EL or EC character, if defined, to be sent to remote user; treated same as NVT characters, i.e., converted to/from NVT model values (see Paragraph 4.4.4); no editing is performed within the network

4. XASCII Shift-out (SO) and Shift-in (SI) - causes user data between SO and SI characters to be sent in XASCII mode (see Paragraph 3.4); meaningful to TMP only if XASCII option has been negotiated (see Paragraphs 4.4.5 and 5.3.3); SO and SI characters will not be recognized if XASCII mode has not been negotiated, i.e., they will be sent as data

5. Go-Ahead (GA) - causes GA character to be sent to remote user, if appropriate, indicating initiator of GA is ready to receive data; meaningful to TMP only if go-ahead option has been negotiated; GA will be sent as data if go-ahead option has not been negotiated (see Paragraphs 4.4.6 and 5.3.5)

6. Interrupt Function Characters (IF1 thru IF19) - cause various interrupt or interrupt/flush functions to be performed by local and/or remote TMP/TCP (see Paragraph 4.4.7)

3.3 RECORD VERSUS STREAM PACKAGING MODES

Record and stream modes refer to the way in which user data is packaged to be sent across the network. Scanning of user data is the same for both packaging modes. In addition, some characters will cause THP records to be sent across the network regardless of scanning mode. The difference is in scanning of data from the network.

A THP record is the unit of information through which THPs communicate (see Appendix G). In record mode all information, user text and THP-THP control, is packaged in THP records. In stream mode only THP-THP control information is packaged in records. Each record type is defined in Appendix A; each has a known length, (or a length field, if variable), and format (see Appendix A). A THP, receiving records from the network can process each record, based on type, without additional scanning of the data stream. A record type, out of the currently defined set of THP-THP records defined in Appendix A, must be in the format: record mark, type, length (16 bits), and parameters (as required). In this way, THPs not implementing the function represented by the record may "skip over" the record (ignore it) based on the length field. It should be noted that the length field contains a value representing the number of bytes in the record after the length field. Effectively the value is the length (in bytes) of the parameter field.

As stated, in record mode all user data as well as THP-THP control information is sent in THP records. Each record begins with a record mark, the value of which is out of the ASCII character range, followed by the record type. THP can process each record, performing required actions, based on the record type with no additional character scanning.

In stream mode user text is sent anywhere within the stream, without record delineations. NVT characters (CR, LF, FF, HT, VT, BEL, BS, NUL) will be converted to NVT model values, included in this text stream, and not sent as NVT records. If THP-THP control must be sent, an appropriate record is inserted in the stream. THP processes the data stream from the network by scanning for NVT model values as well as the record mark. NVT model values must be converted to the local value and put in the to-user buffer along with other text. Conversion of NVT model values is not done in stream mode, if binary mode is in progress, i.e., all octets except record mark will be passed as data. If a record mark is detected THP goes into record mode to process the control record, returning to stream mode when record processing is complete. Text characters are accumulated in to-user buffers and released as required by the user (see Paragraph 3.6).

THPs must implement both modes for from-network processing, but may optionally implement stream mode for packaging user data to the network. AUTODIN II-standard default for packaging user data will be record mode, but this may be changed by the user via the mode command (see Paragraphs 4.1 and 4.3.6). The user would instruct THP to begin packaging his data in either record or stream mode. The user may not request that record or stream mode be used for data coming toward him, i.e., from the network (remote user). To inform the remote THP that data will be packaged in either mode, THP sends a Set Mode record. This is not a request but an announcement of the new mode. If, however, THP-1 desires that THP-2 begin packaging user-2's data in a certain mode, THP-1 would send a Request Mode record. This is a request and although option negotiation is not used the request may be refused. THP-2 would acknowledge or refuse the request by returning a Set Mode record describing the packaging mode which will be used.

3.4 SCANNING USER DATA

There are four modes for scanning user-to-network data in TMP: NVT, XASCII, binary, and transparent. For all modes there are only certain characters which are recognized and which cause special processing in TMP. For each mode there exists a subset of these characters and processes. Essentially, that is the difference between scanning modes: the characters recognized and the processing they invoke. Table 2 summarizes the characters which will be recognized and the processing required by each character for each mode. The scanning of user data is independent of the packaging mode (record versus stream) discussed in Paragraph 3.3.

NVT mode is the known default for processing user-to-network data and is that processing required to format user data in NVT model format (see Paragraph 3.2). This formatting primarily concerns the characters: CR, LF, FF, HT, VT, BS, BEL, and NUL. Each of these characters is converted to an NVT record (see Appendix A) for record mode or an NVT model value for stream mode. The NVT model value is the same as the NVT record type value. The remaining characters of Table 2, data transmission control characters, are processed in one of two ways while in NVT mode:

1. If preceded in the user's data stream by a prefix character, these characters are not converted or acted upon but are passed as data to the network.

2. If detected in the user's data stream and not preceded by a prefix character, the appropriate action is performed, e.g., convert to TMP control record, flush or interrupt function, packet release, or enter command mode (see Paragraph 4.4).

While in NVT mode, all data characters are sent as 7-bit ASCII, i.e., the parity bit will be stripped by TMP before interpretation of the character. If the character requires special processing, that is done; if the character is a nonspecial (data) character, the 7-bit code is transferred to the TMP letter to be sent to the network. The only exception to the NVT mode procedure occurs for the NVT model characters, CR, LF, FF, VT, and HT, the dispositions of which may be changed via option negotiation (see Paragraph 5.3.9).

TABLE 2. TMP DATA SCANNING MATRIX

CHARACTER	NVT		BINARY		XASCII	TRANSPARENT	
	P	NP	P	NP		P	NP
carriage return (CR); linefeed (LF); formfeed (FF); vertical tab (VT); horizontal tab (HT); bell (BEL); backspace (BS); null (NUL)	I	A	I	D	D	I	A
IP1/11; IP6/16; IP9/19	D	B,E	B,E	D	D	B,E	D
IP2/12; IP3/13; IP4/14; IP5/15; IP7/17; IP8/18	D	B,E,F	B,E,F	D	D	B,E,F	D
IP10	D	G	G	D	D	G	D
are-you-there? (AYT)	D	B,E	B,E	D	D	B,E	D
erase character (EC); erase line (EL)	D	B	B	D	D	B	D
go-ahead (GA) (if go-ahead option has been negotiated)	D	B,E	B,E	D	D	D	B,E
XASCII shift-out (SO) (if XASCII option has been negotiated)	I	J	I	D	D	I	J
XASCII shift-in (SI) (if XASCII option has been negotiated)	I	D	I	D	K	I	D
GA; SO; SI (if corresponding option has not been negotiated)	I	D	I	D	D	I	D
send now (SN)	D	E	E	D	D	E	D
prefix (PC)	D	C	D	C	D	D	C

LEGEND:
- A - converted to NVT record (record mode) or NVT value (stream mode)
- B - converted to TMP control record
- C - TMP command or control character expected to follow
- D - passed as data (not recognized as special character)
- E - acts as packet release mechanism
- F - causes forward or reverse flush of data
- G - causes out-of-band TCP function
- I - invalid sequence
- J - activates XASCII data scanning mode
- K - terminates XASCII data scanning mode
- P - when preceded by prefix character
- NP - when not preceded by prefix character

The prefix character has a special meaning for NVT scanning mode, as it does for transparent and binary modes. (In XASCII mode the prefix character is not recognized, but rather is passed as data). The prefix character alerts TMP that the character(s) following it requires special handling. Essentially TMP enters command mode. Section 4 gives a detailed explanation of TMP's processing for user commands, however, three points should be made clear here as they deal with data scanning:

1. If a prefix character is detected in the user's data stream all rules of normal TMP command entry apply (see Paragraph 4.1), i.e., character(s) following the prefix character are expected to be a TMP command

2. Command processing will ensue based on the character(s) after the prefix and the current data scanning mode, e.g., some characters cause different processing in transparent mode than they do in NVT mode

3. If the character sequence is not valid, i.e., not one of the connection control commands, NVT control commands, or data transmission control characters, TMP will perform invalid command processing (see Paragraph 4.1)

Transparent mode is started and stopped via the transparent command (see Paragraph 4.3.4) and is not negotiated with the remote TMP. The difference between NVT and transparent modes is in processing for data transmission control characters. Processing for NVT model characters, such as CR, LF, FF, VT, HT, BEL, BS, and NUL, is the same as for NVT mode. Processing for non-special characters is also the same as for NVT mode: the parity bit is stripped, forming a 7-bit ASCII value. Transparent mode, essentially, allows the user to send data transmission control characters as data, inhibiting TMP recognition. This capability is available in NVT mode by preceding the character with a prefix character. However, to send several of these characters using this procedure would be cumbersome. Transparent mode allows the user to send several of these special characters without each preceding prefix, while processing for normal NVT format continues. As can be seen in Table 2, to invoke the normal NVT mode processing for any special character while in transparent mode, the user simply precedes the character with a prefix character, the opposite of NVT mode procedure. All TMP commands are available while in transparent mode.

Binary mode allows data to flow on the connection without any conversion or special treatment for any characters. The

parity bit is not stripped as in NVT and transparent modes, i.e., the entire 8 bits are sent as data. CCU/TAC TMPs will always package the binary data in Data records, even if the current packaging mode is stream. CCU/TAC TMPs will, however, process binary mode data in stream mode, if received from the network. It should be noted that the scanning for NVT values, normally taking place when receiving in stream mode, will not be done. While in binary mode, receiving in stream mode, all characters (octets) will be passed as is to the user.

Binary mode is entered automatically if the local and remote users' characteristics match upon opening the connection (see Paragraph 5.3.1) or by user command (see Paragraph 4.3.5) and subsequent option negotiation (see Paragraph 5.3.2). Binary mode must be negotiated each time it is entered and will continue until a request to exit binary mode is processed via option negotiation. While in binary mode TMP scans user-to-network data for only the prefix character. If one is detected the subsequent character sequence is expected to be a TMP command or data transmission control character. Connection control commands are processed normally (see Paragraph 4.2). Most NVT control commands are allowed while in binary mode, but changes requested via these commands may not take effect until the connection has returned to NVT mode for the side of the connection affected by the change. The transparent, mode, and move commands are not valid if the connection is in binary mode on the requesting user's send (to network) side. Control characters are processed as usual, but will not be detected unless preceded by prefix character. Other characters, such as NVT model CR, LF, etc., are not recognized or converted, but are passed in the data stream.

XASCII mode allows the user to send data on the connection which will not be processed in any way except to be packaged in XASCII records. That is, all characters (octets) will be passed with no conversion, special processing, nor stripping of the parity bit. The octet, as received from the user, will be packaged in the XASCII record for transmission to the network. The ability to enter XASCII mode must be agreed upon via option negotiation (see Paragraphs 4.3.5 and 5.3.3). Once that agreement is made, XASCII mode may be entered at anytime. TMP enters XASCII mode when an XASCII shift-out (SO) character is detected in the user data stream. From that point until an XASCII shift-in (SI) character is detected in the user data stream, TMP packages the user text in XASCII records. Only the SI will be recognized in XASCII mode. The prefix character and, therefore, TMP commands and special characters, will not be recognized in XASCII mode. The SO and SI characters are not included in XASCII records. Values for these characters must be inserted by the remote TMP, as required by his user. It is the

responsibility of the receiving THP to prevent premature termination of XASCII mode for his user, i.e., premature insertion of SI character. This could occur if an SI were inserted after each XASCII record. Because THP letter size is limited to maximum segment size and THP records are not split between two letters, it may be necessary to package an XASCII session in more than one XASCII record. The session should not be considered to be over, requiring insertion of the local SI character, until the subsequent record is received and it is not an XASCII record.

3.5 PROCESSING NETWORK DATA

Because most of the work is done by THP in packaging user-to-network data, processing data from the network is simplified. THP reacts to each record type without particular regard to the scanning mode used in packaging the data. That is, each record type requires certain procedures. In stream mode (see Paragraph 3.3), of course, not all of the network-to-user stream is in records, so some variation is required. Essentially, in stream mode THP must scan every character in the network-to-user data stream searching for NVT model values, as well as THP record marks. When a record mark is detected in stream mode the record is processed as if the connection were in record mode, returning to stream mode when the record processing is complete. Essentially three types of network-to-user data must be processed:

1. NVT model values (in stream mode) or NVT records (in record mode) for CR, LF, FF, HT, VT, BEL, BS, NUL - if detected, local values are substituted for these NVT model values in the current to-user buffer; an exception is made if in binary mode for network-to-user data; if in stream mode, NVT model values are merely passed as data to the user

2. THP-THP control records - record is processed as if in record mode; THP-THP control records include set mode, request mode, status, status reply, data mark, option negotiation, RCTE control, interrupt functions (1-9 and 11-19), AYT, EC, EL, GA; XASCII, and data records if network-to-user side of connection is in binary mode

3. Interrupt function 10 - this interrupt function is communicated to the remote THP via an out-of-band TCP control command and TCP-to-THP Interrupt Return event (see Paragraph 4.4.7)

If an unknown record, i.e., type (second byte) is not defined in Appendix A, is received from the network, the record will be ignored. THP assumes that there is a length field and that the length field contains a value representing the number of bytes in the record after the length field. The third and fourth bytes of the record are assumed to be the low- and high-order eight bits, respectively, of the length field. THP will increment past the unknown record and continue processing as normal. As discussed in Paragraph 2.3.3, there are certain records which, if received under certain circumstances create protocol errors. However, unknown record types are simply ignored (bypassed based on the length given in the record).

3.6 RELEASE MECHANISMS

TMP control records and user text, whether sent within the network in record or stream mode, are packaged in TMP letters (see Appendix G). These letters are sent to TCP for transmission over the network. There may be one or many TMP records in a letter but partial records are not permitted or supported by CCU/TAC TMPs or TCPs. A letter is sent to the network under certain circumstances, which are either directly or indirectly created by the user. Of the list below, the user may select one of the first four to be the primary packet release mechanism via the packet release command (see Paragraph 4.3.3). Only one of the first three may be active at a time. That one will be the primary method which is used if none of the other mechanisms listed occurs first. These other mechanisms are under user control only indirectly in that they are predicated on user input. All of these mechanisms cause the current letter to be sent to the network. If there is no current letter, the record or function generated by the mechanism is sent immediately to the network. In addition, the send now and full segment mechanisms override whatever mechanism the user may have specified (of the first three). The following conditions cause a letter to be sent to the network:

1. As received - release upon receipt of a unit of user data from HSI or TM (From User event), defined by the Host-CCU or terminal-TAC link protocol transmission unit (see Appendix G); must be specified by the user as packet release mechanism (see Paragraph 4.3.3)

2. end-of-line - release upon detection of local end-of-line sequence (in user's data stream) such as CR/LF or newline (NL) character; must be specified by user as packet release mechanism; this mechanism is not "active" if user-to-network data is in binary mode (see Paragraph 4.3.3)

3. number of characters - release upon accumulation of a specified number of characters; the number of characters must be specified by user as packet release mechanism and may not be greater than 584, the maximum number of data characters which can be sent in an AUTODIN II packet allowing for TMP overhead characters (see Paragraph 4.3.3)

4. send now - release upon detection of send now (SN) character; overrides all other packet release mechanisms; in binary mode SN must be preceded by a prefix character (see Paragraph 4.4.2)

5. full segment - release when letter size has reached

maximum segment size (588 octets) or the users maximum buffer size, based on value generated in UPG and changed by packet release command; overrides all other packet release mechanisms

6. are-you-there? - release upon detection of AYT character (see Paragraph 4.4.3)

7. RCTE - release upon detection of RCTE command (CCU only); RCTE option must have been negotiated (see Paragraphs 4.3.8 and 5.3.4)

8. interrupt function - release upon detection of interrupt function character (IF1-19); letter is released for all of these, including IF10 which does not require a TNP-TNP control record (see Paragraph 4.4.7)

9. go-ahead - release upon detection of go-ahead character; go-ahead option must have been negotiated (see Paragraph 5.3.5)

Network-to-user data will be released to the user in a logical unit, as required by that user based on protocol or other user requirements. This logical unit is established at system generation time and is not changeable. Only one of the following may be specified as the text release mechanism for the user:

1. upon accumulation of certain number of characters

2. as received from network, i.e., TNP letter

3. upon detection of NVT end-of-line sequence (CR/LF); only in NVT mode

4. character-at-a-time (special case of number 1)

SECTION 4. USER COMMANDS

4.1 GENERAL INFORMATION

TMP commands are provided to allow the user some control over his connection and the processing of his data. Commands are grouped into three categories: (1) connection control commands, (2) NVT control commands, and (3) data transmission control characters. These commands are summarized in Tables 3 through 5 (see references in paragraphs for each category). Each command or category of commands has a special meaning and considerations or restrictions on use, however, the following general rules apply to all commands:

1. commands may be entered anywhere in user-to-network data stream

2. TMP prefix character (PC) has special meaning for command processing

   1. PC must precede each (or the first in a series of) connection control and NVT control commands for them to be acted upon; it is considered part of the command sequence

   2. in NVT mode PC must not precede data transmission control characters (IF1 thru IF19, AYT, EC, EL, GA, SN, PC) for them to be acted upon; these characters will be sent as data if preceded by PC in NVT mode

   3. in binary or transparent mode PC must precede data transmission control characters (IF1 thru IF19, AYT, EC, EL, GA, SN) for them to be acted upon; these will be sent as data if not preceded by prefix in these modes; GA will not be affected by transparent mode, i.e., NVT rules hold true for GA character

   4. PC and, therefore, TMP commands will not be recognized if user-to-network scanning mode is XASCII; only SI character is recognized in XASCII mode

   5. the second consecutive PC will always be sent as data, regardless of the scanning mode

   6. because PC is a special character, other characters (IF1 thru IF19, AYT, EC, EL, GA, SN) may not be set to a value equal to the current value of PC; in turn, PC may not be set to the

current value of any of these characters (see Paragraph 4.3.2)

3. If parameters are required they will be positional and multiple parameters will be separated by commas; (e.g., <PC>CMD param1,param2,param3.)

4. If one parameter may be omitted in a command sequence, the separator (comma) must be included (e.g., <PC>CMD param1,,param3.)

5. TMP commands will not be recognized in XASCII mode

6. More than one connection control or NVT control command may be entered in a command sequence but must be separated by semicolons; in this case PC is dropped from the second and subsequent commands, (e.g., <PC>CMD param,param;CMD2;CMD3 param.)

7. Connection control and NVT control command strings must be terminated by a period; (e.g., <PC>CMD param,param.)

8. Any connection control or NVT control command may be set as unavailable for use by a particular user; this indicator is set administratively and included in the system generation, therefore, it may not be changed; (the restriction is more feasible for NVT control rather than connection control commands, but is available for all commands)

9. Some commands have restricted usage, e.g., only during an active connection; these restrictions are enumerated in paragraphs describing the individual commands

10. The period and semicolon have special meaning to command processing, therefore, other TMP special characters (CR, LF, FF, HT, VT, BS, BEL, NUL, PC, AYT, EC, EL, GA, SO, SI, SN, IF1-19) may not be set (via set command) to a value equal to the period or semicolon

If a command sequence is determined to be in error (invalid command, illegal usage, incomplete or invalid command parameters, etc.), the user will be notified of the error via a TMP output message (see Paragraph 2.1.4 and Appendix D). In addition, that command (all characters following the PC up to and including the terminating character) will be discarded. If an errored command is terminated by a semicolon, the next command in the sequence will be processed. If an errored command is terminated by a period, subsequent char-

acters will be considered data to be processed for the network, i.e., TMP will leave command mode. If a period or semicolon is not found within the next 62 characters after the start of the command (prefix or semicolon), TMP will terminate the search, will discard the entire 62 characters, and will leave command mode. The errored command sequence, essentially the discarded data, will be included in the invalid command notice to the user.

4.2 CONNECTION CONTROL COMMANDS

Connection control commands are those which allow the user to begin (open or listen) or end (close or abort) a connection. Table 3 summarizes these commands, giving format for each. In addition to these four commands there are two automatic features for connection control: auto-listen and auto-open. These are discussed in Paragraph 4.2.3. All connection control commands are available to all users unless marked unavailable administratively (see Paragraph 4.1).

4.2.1 Open Command

The open command is interpreted as a request to establish a connection with a specific remote user (see Paragraph 2.2). The command is allowed only if the connection state is closed or listening. This command overrides a previous listen command or auto-listen function if the connection is still in a listening state, and overrides an auto-open if the open command sequence is found in first buffer (after link protocol has been established or a connection has closed) from the user. Required parameters are:

1. precedence - one of 16 codes for AUTODIN II and authorized for this user; codes are: A and B (not available for use by subscribers), C thru N, S, and T (see References 5 and 9)

2. security - one of eight codes for AUTODIN II and authorized for this user would be inserted in a four-character field; security codes are: PROG (not available for use by subscribers), MMMM, TSEC, SECR, CONF, REST, EFTO, and UNCL (see Reference 5)

3. transmission control code (TCC) - one of the subset of 512 AUTODIN II alphabetic trigraphs authorized for this user

4. destination subscriber address - up-to-five decimal characters in range from 0 to 65535

5. destination port ID - this is actually three parameters (separated in the command sequence by commas) but the three are considered as one logical unit; first, user ID type - specifies either static (S) or dynamic (D), presumably used by the host; second, user ID - up-to-four decimal characters in range from 0 to 2047, used by TCP (see Reference 9) in checking the S/P/T authorization for the user; and third, function suffix - up-to-two decimal

characters in range from 3 to 15, presumed to be used by the host in determining the process or function utilizing the connection; the destination port ID field must be specified (even if it is zero as it would be for a TAC terminal user); this requirement exists for the CCU/TAC THP command processor and is not required for correct protocol operation or for communication with a TAC or CCU unless the CCU-connected host so dictates

6. source port id - this is in the same format as for the destination port ID and is not required if source subscriber is TAC terminal user; it must, however, be specified (even if it is zero), if source subscriber is connected to a CCU; this requirement exists for the CCU THP command processor and is not required for correct protocol operation

7. dial-in user ID - up-to-five character user ID for a TAC dial-in subscriber; required for TAC dial-in subscribers only.

8. dial-in password - up-to-seven character password associated with dial-in user ID; required for TAC dial-in subscribers only

9. terminal type - code designating type of user profile information to be used by a dial-in subscriber; optional for TAC dial-in subscribers only

TABLE 3. THP CONNECTION CONTROL COMMANDS

COMMAND DESCRIPTION	COMMAND SYNTAX	PARAGRAPH REFERENCE
Open command - allows user to request that connection be established according to parameters given in command sequence; dial-in users on the TAC will be required to furnish user identification and terminal type as well as the normal open parameters	OPN p,ssss,tcc,dsadr,$\underbrace{\text{u,usid,fc}}_{\text{destination}}$/$\underbrace{\begin{matrix}\text{u,usid,fc}\\ \text{usrid,pass,tp}\end{matrix}}_{\text{source}}$ p - precedence (A-N,S,T) ssss - security (PROG,MMMM,TSEC,SECR,CONF,REST,EFTO,UNCL) tcc - transmission control code; alphabetic trigraph dsadr - destination subscriber address; up-to-five decimal characters (0-65535) u - user ID type; specifies either static (S) or dynamic (D) usid - user ID; up-to-four decimal characters (0-2047) fc - function suffix; up-to-two decimal characters (0-15) usrid - dial-in user ID; up-to-five character user ID for dial-in users on TAC only pass - dial-in password; up-to-seven character password associated with dial-in user ID for dial-in users on TAC only tp - terminal type; optional two-character terminal type indicator for dial-in users on TAC only NOTE: parameter sequence "u,usid,fc" is not specified if corresponding user (destination or source) is a terminal user connected to TAC, but must be specified if corresponding user is connected to a CCU NOTE: "tp" codes are to be assigned; users may contact DCA Code 450 for current assignments	4.2.1

Page 1 of 2.

Revision ., July 4, 1979

TABLE 3. THP CONNECTION CONTROL COMMANDS

COMMAND DESCRIPTION	COMMAND SYNTAX	PARAGRAPH REFERENCE
Listen command - allows user to indicate that he is ready for an incoming connection to be established based on parameters given in command sequence; all parameters may be omitted, except the source "u,usid,fc" field for a requester that is connected to a CCU	LIS p,ssss,tcc,dsadr,$\underbrace{\text{u,usid,fc}}_{\text{destination}}$/$\underbrace{\text{u,usid,fc}}_{\text{source}}$ NOTE: parameter descriptions are same as for Open command except that for users connected to a TAC listen command sequence ends after destination "fc," replacing "/" with the command terminator (period or semicolon)	4.2.2
Abort command - allows user to request immediate termination of connection without regard to delivery of data currently in network	ABO	4.2.5
Close command - allows user to request that connection be terminated when data entered by requester (prior to Close command) is delivered to destination user	CLS	4.2.4

Page 2 of 2.

4.2.2 Listen Command

The listen command notifies THP that the user is ready for an incoming connection. The command is allowed only if the connection state is closed or listening. This command overrides a previous listen command or an auto-listen function if the connection is still in the listening state, and overrides an auto-open function if the listen command sequence is found in the first buffer (after link protocol has been established or a connection has closed) from the user. Parameters for this command are identical to those for an open command (see Paragraph 4.2.1). For a listen command, however, some parameters may be omitted. The only required parameter is the source port ID when the command is entered by a CCU user. If no (other) parameters are given the connection may be established with any remote user (providing he meets security, precedence, and TCC requirements for this user). The more parameters specified for a listen, the smaller the field of prospective users with which a connection may be established.

4.2.3 Automatic Features

There are two automatic features available to CCU and TAC users: auto-open and auto-listen. These features are provided especially for those users which do not wish to (or cannot) enter THP commands, but they are available to any user. A user may be designated as either auto-open or auto-listen type or as both. Parameters required (see Paragraphs 4.2.1 and 4.2.2) will be incorporated at system generation time into the user's profile. One set of parameters is provided, i.e., auto-open and auto-listen parameters must be the same. However, a subset may be designated for use for the auto-listen function, i.e., only certain parameters would be specified for the auto-listen request. In any case, manual open and listen commands may be entered to override the automatic feature.

An auto-listen function begins when the link (channel) is activated or when a connection is closed but the link is still active for an auto-listen type user. The former will be detected by HSI or TH when the link protocol has been initialized or the line is brought to a ready state, as appropriate for the channel. The latter is recognized by THP itself. When an auto-listen function is warranted, THP will proceed as if a listen command had been entered, using parameters specified in the user profile. Auto-listen, just as any listen, may be cancelled by a manual open or listen command.

An auto-open function begins when user-to-network data is received from an auto-open type user and connection state is closed or listening. THP will proceed as if an open command had been entered, using parameters specified in the user profile. Auto-open may be superceded by an open command present in the first buffer (after link protocol has been established or a connection has closed) from the user.

If a user is both an auto-listen and an auto-open type, a listen will be outstanding from the time the link is activated until data for the network is received from the user. At that time the auto-open function will begin, unless, of course, an overriding listen or open command sequence is detected in that first buffer.

4.2.4 Close Command

The close command instructs THP to close an active connection as soon as all data entered by the requester prior to the close command has been delivered to the destination user. The connection must be active for the close request to take effect. If a connection has not been established or requested (active or opening) the close command will not be valid. There are no parameters required for a close command.

4.2.5 Abort Command

The abort command instructs THP to close an active connection immediately, without regard to delivery of data to the destination user. The abort command will be valid unless the state is already aborting.

4.3 NVT CONTROL COMMANDS

NVT control commands are those which allow the user to request that certain actions be accomplished in regard to his connection or user profile. Table 4 summarizes these commands, giving the format of each. Some NVT control commands may be administratively restricted from use for some users. This restriction will be controlled via the user profile and will be set by system generation. There are no specific examples of why these restrictions may be imposed other than that a user would not require or be authorized to use the feature. For example, there may be need for only one CCU user (a "control" user) to enter a move command (see Paragraph 4.3.7) requesting that his connection be moved to another user on the same CCU. In this situation, all other CCU users would be prohibited from using the move command. In addition, a user authorized to use the move command would probably not be allowed to use the option commands, as he would be acting merely as a log-in facility for other users on the MCCU. There would seem to be little purpose in administratively restricting the status command (see Paragraph 4.3.1). However, if a user is one which receives no TMP output messages (see Paragraph 2.1.4), it may be appropriate to restrict use of the status command for that user. Of course, TMP output message format may be changed via the set command (see Paragraph 4.3.2), so it is not clear that the restriction is warranted in this case. Requirements for each user must be considered separately.

4.3.1 Status Command

This command allows the user to request his current status in relation to the network, e.g., "connection established." Status of the remote user (host process or terminal) must be acquired via the are-you-there? feature (see Paragraph 4.4.3). In processing the status command, TMP will first query local TCP (via Status event) and then remote TMP (Status record), if required. This procedure will ensure that the network and remote access facility (CCU, TAC, etc.) are functioning. There are no parameters for this command. There are no restrictions to the use of this command, i.e., it may be entered at any time. See Appendix D for other TMP output messages which may be generated in response to a status command.

TABLE 4, THP NVT CONTROL COMMANDS

COMMAND DESCRIPTION	COMMAND SYNTAX	PARAGRAPH REFERENCE
Status command - allows user to request current status of connection as relates to network (destination user's status may be obtained via are-you-there? data transmission control character)	STS	4.3.1
Set command - allows user to change certain characteristics of his user profile	SET ccc,ppp ccc - characteristic to be changed ppp - additional parameters, as required NOTE: Set command parameters are enumerated in Table 5	4.3.2
Packet Release command - allows user to change mechanism to be used in releasing text to network, as specified by parameter in command sequence	PKR $\left\{\begin{matrix} \text{EOL} \\ \text{nnn} \\ \text{ASR} \\ \text{SNW} \end{matrix}\right\}$ EOL - upon detection of end-of-line sequence nnn - upon accumulation of specified number of characters (up to maximum allowed, 584 characters) ASR - as received from ESI/TM (unit of transmission) SNW - upon detection of send now character	4.3.3
Transparent command - allows user to instruct THP to enter or leave transparent mode for scanning data coming from user going toward network	TRP $\left\{\begin{matrix} \text{ON} \\ \text{OFF} \end{matrix}\right\}$	4.3.4
Mode command - allows user to instruct THP as to packaging mode (record or stream) for data going toward network	MOD $\left\{\begin{matrix} \text{S} \\ \text{R} \end{matrix}\right\}$ S - stream mode R - record mode	4.3.6

Page 1 of 4.

TABLE 4. TMP NVT CONTROL COMMANDS (CONT'D)

COMMAND DESCRIPTION	COMMAND SYNTAX	PARAGRAPH REFERENCE
Remote Control of Transmission and Echo (RCTE) command - allows user (CCU only) to control hardware echo and packet release mechanism, as specified by parameters in command sequence, for Mode IIA terminal at remote end of connection; RCTE option must have been negotiated successfully	RCT {OFF, ON}, {EOL, nnn, ASR, SNW} OFF/ON - refer to desired condition of Mode IIA hardware echo EOL/nnn/ASR/SNW - refer to new packet release mechanism to be used for terminal user's data NOTE: either or both of these parameter categories may be specified	4.3.8
Echo command - allows Mode IIA terminal user (TAC only) to control local hardware echo; command is not valid if RCTE option has been negotiated successfully	ECH {OFF, ON} OFF/ON - refer to desired condition of Mode IIA hardware echo	4.3.10
Full Duplex command - equivalent to ECH ON (echo on) command for Mode IIA terminal user (TAC only)	FDX	4.3.10
Half Duplex command - equivalent to ECH OFF (echo off) command for Mode IIA terminal user (TAC only)	HDX	4.3.10
Move Connection command - allows user (MCCU only) to request that his active connection be moved to another local socket on the same MCCU, specified by parameters in command sequence	MOV u,usid,fc u - user ID type; specifies either static (S) or dynamic (D) usid - user ID; up-to-four decimal characters (0 to 2047) fc - function suffix; up-to-two decimal characters (0 to 15) NOTE: parameter sequence "u,usid,fc" specify receiving local socket	4.3.7

Page 2 of 4.

TABLE 4. NVT CONTROL COMMANDS (CONT'D)

COMMAND DESCRIPTION	COMMAND SYNTAX	PARAGRAPH REFERENCE
Option commands - allow user to request that certain alternatives to normal NVT processing be started or stopped; requests result in option negotiation processing between the two THPs, as required; user specifies which THP will perform option, as well as which side of connection is concerned (his send or receive side), by the command he uses; parameters in command sequence specify which option and any corresponding values	DOS / DNS / DOR / DNR / WLS / WNS / WLR / WNR } option,parameters NOTE: options and corresponding parameters are enumerated in Table 6 NOTE: command used specifies details of how option will be processed and for which side of connection 1st letter { D - requester's send side; W - requester's receive side 2nd letter { O - start option; L - start option; N - stop option 3rd letter { R - THP receiving data; S - THP sending data NOTE: certain options which require that both THPs do processing, e.g., binary mode, XASCII, ECTE, and go-ahead, must be requested with either DOR, DNR, WLR, or WNR, as appropriate NOTE: CCU/TAC THPs will not allow a request that would result in THP's performing non-NVT processing on the send side of the connection, i.e., DOS and DNS commands are not supported	4.3.5

Page 3 of 4.

TABLE 4, NVT CONTROL COMMANDS (CONT'D)

COMMAND DESCRIPTION	COMMAND SYNTAX	PARAGRAPH REFERENCE
Time command - allow user to request that the connection inactivity timer be turned off or on; it should be noted that a timer will always run during option negotiation, but the connection i- ctivity timer is controlled by the user for a other transmissions	TIM {ON / OFF}	4.3.9

Page 4 of 4.

4.3.2 Set Command

The set command allows the user to change certain characteristics within his user profile. Characteristics which affect network processing (user-to-network or network-to-user data processing) and would require notifying the remote TMP of change must be changed via the option commands (see Paragraph 4.3.5). Some characteristics are changed by special commands, such as, the packet release mechanism (see Paragraph 4.3.3), scanning mode (see Paragraph 4.3.4), packaging modes (see Paragraph 4.3.6), connection inactivity timer (see Paragraph 4.3.9), or echo feature (see Paragraph 4.3.17). Other characteristics, which relate more to the physical aspects or NVT default values for the user, must be changed via the set command. These changes require no communication with remote TMP. "Physical" aspects are those aspects which are actual hardware restrictions or settings for a certain characteristic. For example, physical paper width is the number of characters which can be printed on a line before the hardware will "fold" the line or "pile up" at the end-of-line. Default values for receive characteristics are the restrictions imposed by the user. These are used to restore the "default" condition when an option is stopped. That is, if an option is invoked for handling some characteristic, values used in processing user data will be changed (dynamic values). When the option is terminated, there must be an NVT default to which TMP can fall back. The user establishes his default values, at system generation time or via the set command. The set command parameters, restrictions, and formats required to change the various characteristics are shown in Table 5 and summarized here:

1. paper length - both physical and default length of paper or screen

2. paper width - both physical and default width of paper or screen

3. hardware horizontal tab settings - for terminals with hardware settable horizontal tabs

4. default horizontal tab settings

5. hardware vertical tab settings - for terminals with hardware settable vertical tabs

6. default vertical tab settings

7. special character values - IF1 thru IF19, AYT, GA, SO, SI, PC, BEL, BS, NUL, CR, LF, FF, HT, VT, SN, EC, and EL

8. TNP output message format - format for TNP informational messages (verbose, concise, or none)

9. options to be negotiated - indicate to TNP that option negotiation should or should not be allowed (for local or remote user) for particular options (binary mode, RCTE, go-ahead, XASCII mode, line width, page size, horizontal tab settings, and vertical tab settings)

TABLE 5. SET COMMAND PARAMETERS
****

PARAMETER DESCRIPTION	COMMAND SYNTAX
hardware paper length - value assigned to this parameter represents the physical paper length, i.e., number of lines which may be printed on user's "terminal", value is used to determine if option request (from remote TMP) for page size or vertical tab settings is feasible	SET HPL,value value - number of lines per page (1 - 250)
default paper length - value assigned to this parameter represents the page size, i.e., number of lines on a page; value is used to restore the dynamic page size for this user, which is the number of lines per page to be used in formatting data going to user from network; dynamic page size is restored when a page size option is terminated (returned to default condition) via option negotiation	SET DPL,value value - number of lines per page (1 - 250)
hardware paper width - value assigned to this parameter represents the physical paper width, i.e., number of characters which may be printed per line on user's "terminal" value is used to determine if option request (from remote TMP) for line width or horizontal tab settings is feasible	SET HPW,value value - number of characters per line (1 - 250)
default paper width - value assigned to this parameter represents the line width, i.e., number of characters on a line; value is used to restore the dynamic line width for this user, which is the number of characters per line to be used in formatting data going to user from network; dynamic line width is restored when a line width option is terminated (returned to default condition) via option negotiation	SET DPW,value value - number of characters per line (1 - 250)

Page 1 of 5.

TABLE 5. SET COMMAND PARAMETERS (CONT'D)

PARAMETER DESCRIPTION	COMMAND SYNTAX
<u>hardware horizontal tab settings</u> - value(s) assigned to this parameter represent horizontal tab settings of user's terminal such that if a horizontal tab character were sent to terminal, the hardware would perform the tab function; value(s) are used to perform characteristics checking during open processing and to determine if simulation is required for horizontal tab function to user	SET HHT,$\begin{Bmatrix} *n \\ n,n,n,n,n,n \end{Bmatrix}$,$\begin{Bmatrix} S \\ R \end{Bmatrix}$ *n - tab settings at every nth column beginning with column n n,n,n,n,n,n - tab settings at designated columns, maximum of six tabs within bounds of hardware paper width S - user's send side (these values used for characteristics checking only) R - user's receive side
<u>default horizontal tab settings</u> - value(s) assigned to this parameter represent horizontal tab settings to be used to restore dynamic horizontal tab settings to be used in simulating horizontal tab function, as required, in formatting data going to user from network; dynamic horizontal tab settings are restored when a horizontal tab stops option is terminated (returned to default condition) via option negotiation	SET DHT,$\begin{Bmatrix} *n \\ n,n,n,n,n,n \end{Bmatrix}$ *n - tab settings at every nth column beginning with column n n,n,n,n,n,n - tab settings at designated columns, maximum of six tabs within bounds of hardware paper width
<u>hardware vertical tab settings</u> - value(s) assigned to this parameter represent vertical tab settings of user's terminal such that if a vertical tab character were sent to terminal the hardware would perform the tab function; value(s) are used to perform characteristics checking during open processing and to determine if simulation is required for vertical tab function to user	SET HVT,$\begin{Bmatrix} *n \\ n,n,n,n,n,n \end{Bmatrix}$,$\begin{Bmatrix} S \\ R \end{Bmatrix}$ *n - tab settings at every nth line beginning with line n n,n,n,n,n,n - tab settings at designated lines, maximum of six tabs within bounds of hardware paper length S - user's send side (these values used for characteristics checking only) R - user's receive side

Page 2 of 5.

TABLE 5, SET COMMAND PARAMETERS (CONT'D)

PARAMETER DESCRIPTION	COMMAND SYNTAX
default vertical tab settings - value(s) assigned to this parameter represent vertical tab settings to be used to restore dynamic vertical tab settings to be used in simulating vertical tab function, as required, in formatting data going to user from network; dynamic vertical tab settings are restored when a vertical tab stops option is terminated (returned to default condition) via option negotiation	SET DVT, $\begin{Bmatrix} *n \\ n,n,n,n,n,n \end{Bmatrix}$ *n - tab settings at every nth line beginning with line n n,n,n,n,n,n - tab settings at designated lines, maximum of six tabs within bounds of hardware paper length
change prefix character value - allows user to redefine value to be used to scan for prefix character in user's data stream	SET PC,value value - must be valid ASCII, non-upper case alpha character, and may not be assigned value equal to: space, semicolon, period, or current value of any data transmission control character
change interrupt function character value - allows user to redefine value for one of interrupt function characters to be used in scanning user's data stream or in converting to character value for output to user	SET IFn, $\begin{Bmatrix} \text{value} \\ \text{BRK} \end{Bmatrix}, \begin{Bmatrix} S \\ R \end{Bmatrix}$ n - designates which interrupt function character value is to be changed value - must be valid ASCII, non-upper case alpha character, and may not be assigned value equal to: semicolon, period, or current prefix character BRK - indicates that interrupt function will be represented by hardware break signal rather than a character value S - user's send side R - user's receive side NOTE: if "value" is an ASCII space or omitted, the designated interrupt function will not be represented for that side of the connection

Page 3 of 3.

TABLE 5, SET COMMAND PARAMETERS (CONT'D)

PARAMETER DESCRIPTION	COMMAND SYNTAX
change special character value - allows user to redefine value for one of special characters to be used in scanning user's data stream or in converting to character value for output to user	SET char,value,{S/R} char - acronym for character to be assigned new value (AYT, GA, EC, EL, SO, SI, CR, LF, FF, HT, VT, BEL, BS, NUL) value - must be valid ASCII character, and may not be assigned value equal to: semicolon or period; AYT, GA, EC, EL, SO, and SI must be non-upper case alpha character value; AYT, GA, EC, and EL may not be assigned value equal to current prefix character S - user's send side R - user's receive side NOTE: if "value" is an ASCII space or omitted and "S" qualifier is used, designated special character will not be recognized in the user-to-network data stream; if "value" is an ASCII space or omitted and "R" qualifier is used, designated character will be represented by an ASCII space when sent to user; if the user does not wish to receive a special character, he must request that the character be discarded for his receive side via the WLR option command
change send now character value - allows user to redefine value to be used to scan for send now character in user's data stream	SET SN,value value - must be valid ASCII, non-upper case alpha character, and may not be assigned value equal to: semicolon, period, or current prefix character NOTE: if "value" is an ASCII space or omitted, the send now character will not be represented, i.e., there will be no send now character acting as a packet release mechanism

Page 4 of 5.

TABLE 5. SET COMMAND PARAMETERS (CONT'D)

PARAMETER DESCRIPTION	COMMAND SYNTAX
change negotiability of option - allows user to control which options should be negotiated (agreed to) with remote TMP, as specified by parameters in command sequence	SET {BIN / RCT / GOA / XAS / LW / PS / HTL / VTL}, {YES / NO}, {S / R} BIN/RCT/GOA/XAS/LW/PS/HTL/VTL - option for which negotiability is being changed (binary, RCTE, go-ahead, XASCII, line width, page size, horizontal tab settings, and vertical tab settings, respectively) YES - indicates user desires TMP to respond positively (agree to) a valid option request for specified option NO - indicates user desires TMP to respond negatively (do not agree to) any option request for specified option S - user's send side R - user's receive side
change TMP output message format - allows user to specify format of TMP output messages (informational messages to user) via parameter in command sequence	SET MSG, {OFF / VBS / CON} OFF - indicates that user desires no TMP informational messages to be sent to him VBS - indicates that user desires entire message text (prefix,message number,text) to be sent to him CON - indicates that user desires a concise message text (prefix,message number) to be sent to him

Page 5 of 5.

4.3.3 Packet Release Command

The packet release command allows the user to change the packet release method to be used for data being sent from him to the network. If the RCTE option has been negotiated for a TAC terminal, the packet release command is not available to him. In allowing RCTE to be negotiated, the user forfeited control of the packet release mechanism and echo feature for his terminal. There are four alternatives, one of which may be specified by the user via parameters (see Table 4). The alternatives, also discussed in Paragraph 3.6, are:

1. as received - release upon receipt of a unit of user data from HSI or TH (From User event), presumably Host-CCU or terminal-TAC link transmission unit (see Appendix G)

2. end-of-line - release upon detection of local end-of-line sequence such as CR/LF or newline (NL) character; this mechanism is not active when the user-to-network side of the connection is in binary mode

3. number of characters - release upon accumulation of specified number of characters, not greater than 584, the maximum number of data characters which can be sent in an AUTODIN II packet allowing for THP overhead characters

4. send now - release upon detection of send now (SN) character; overrides all other packet release mechanisms; in binary mode SN must be preceded by prefix character to act as a packet release mechanism

4.3.4 Transparent Command

The transparent command allows the user to enter or leave transparent scanning mode for data going to network. Parameters are ON or OFF, as shown in Table 4, and either must be specified. When in transparent mode certain data transmission control characters will not be recognized as they are in NVT mode (see Paragraph 3.4).

4.3.5 Option Commands

Option commands and parameters are summarized in Tables 4 and 6. These commands allow the user to request certain variations to NVT model processing for data flowing on his connection. The request may relate to data going toward or coming from the network. As shown in Table 6, options may be requested of either THP (sender or receiver of data). THP (serving the requester) will build option negotiation records, as required, and send them to the remote THP. The user will be informed of the outcome of negotiations.

Just as any other NVT control command, option commands, may be restricted for use by some users. In these cases, the user would be informed that the command is not available. Certain options may be marked nonnegotiable for either the user's send or receive side. THP will refuse to negotiate (will send negative response) if it receives an option negotiation request from remote THP for one with this restriction. Paragraphs 5.1, 5.2, and 5.3 discuss option negotiation in detail. Negotiability is established at system generation time and may be changed via the set command (see Paragraph 4.3.2).

The requesting user will always be informed of the results of option negotiation, i.e., if the request was accepted or denied. In addition, if a binary, RCTE, go-ahead, or XASCII option is agreed to (started or stopped) the other user (not requesting user) will be informed. This is done because it is felt that these four options have significance beyond the NVT printer and the user should be cognizant of the agreement.

TABLE 6. OPTION COMMAND PARAMETERS

PARAMETER DESCRIPTION	COMMAND SYNTAX
binary mode option - allows user to request that THP attempt to negotiate binary mode for specified side (user's send or receive) of connection; side of connection is indicated in command used; while in binary mode, NVT processing is suspended	{DOR DNR WLR WNR} BIN NOTE: only receive-oriented commands are valid for this option as both THPs will be responsible for processing
RCTE option - allow user to request that THP attempt to negotiate RCTE option for specified side (user's send or receive) of connection; side of connection is indicated in command used; this option allows a CCU user to control echo feature for a Mode IIA terminal user and to control packet release mechanism for any terminal user connected to a TAC	{DOR DNR WLR WNR} RCT NOTE: only receive-oriented commands are valid for this option as both THPs will be responsible for processing; in addition, only WLR and WNR are appropriate for CCU users, and only DOR and DNR are appropriate for TAC users, as these commands correspond to the affected side of the connection for those users
go-ahead option - allows user to request that THP attempt to negotiate go-ahead option for specified side (user's send or receive) of connection; side of connection is indicated in command used	{DOR DNR WLR WNR} GOA NOTE: only receive-oriented commands are valid for this option as both THPs will be responsible for processing
XASCII option - allows user to request that THP attempt to negotiate XASCII option for specified side (user's send or receive) of connection; side of connection is indicated in command used	{DOR DNR WLR WNR} XAS NOTE: only receive-oriented commands are valid for this option as both THPs will be responsible for processing

Page 1 of 4.

TABLE 6. OPTION COMMAND PARAMETERS

PARAMETER DESCRIPTION	COMMAND SYNTAX
<u>line width option</u> - allows user to request that value used for maximum line width on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd LW,value cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) value - line width value; must be within bounds of hardware paper width if request is WLR or WLS; must be within range 1-250 if request is DOR; not specified if request is DNR, WNR, or WNS
<u>page size option</u> - allows user to request that value used for maximum page size on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd PS,value cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) value - page size value; must be within bounds of hardware paper length if request is WLR or WLS; must be within range 1-250 if request is DOR; not specified if request is DNR, WNR, or WNS
<u>horizontal tab stops option</u> - allows user to request that value(s) used for horizontal tab settings on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd HT, { *n / n,n,n,n,n,n } cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) *n - tab settings at every nth column beginning with column n (see note) n,n,n,n,n,n - tab settings at designated columns, maximum of six tabs (see note) NOTE: "n" must be within bounds of hardware paper width if request is WLR or WLS; must be within range 1-250 if request is DOR; not specified if request is DNR, WNR, or WNS

Page 2 of 4.

TABLE 6. OPTION COMMAND PARAMETERS

PARAMETER DESCRIPTION	COMMAND SYNTAX
vertical tab stops option - allows user to request that value(s) used for vertical tab settings on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd VT, {*n / n,n,n,n,n,n} cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) *n - tab settings at every nth line beginning with line n (see note) n,n,n,n,n,n - tab settings at designated lines, maximum of six tabs (see note) NOTE: "n" must be within bounds of hardware paper length if request is WLR or WLS; must be within range 1-250 if request is DOR; not specified if request is DNR, WNR, or WNS
carriage return disposition option - allows user to request that method of handling carriage return on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd CRD,value cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) value - carriage return disposition indicator: 0-250 = insert specified number of nulls following character (for time delays) 254 = discard character 255 = deliver character as received
linefeed disposition option - allows user to request that method of handling linefeed on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd LFD,value cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) value - linefeed disposition indicator: 0-250 = insert specified number of nulls following character (for time delays) 253 = simulate linefeed function 254 = discard character 255 = deliver character as received

Page 3 of 4.

TABLE 6. OPTION COMMAND PARAMETERS

PARAMETER DESCRIPTION	COMMAND SYNTAX
formfeed disposition option - allows user to request that method of handling formfeed on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd FFD,value cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) value - formfeed disposition indicator: 0-250 = insert specified number of nulls following character (for time delays) 252 = replace formfeed with end-of-line function 253 = simulate formfeed function 254 = discard character 255 = deliver character as received
horizontal tab disposition option - allows user to request that method of handling horizontal tab on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd HTD,value cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) value - horizontal tab disposition indicator: 0-250 = insert specified number of nulls following character (for time delays) 253 = simulate horizontal tab function 254 = discard character 255 = deliver character as received
vertical tab disposition option - allows user to request that method of handling vertical tab on specified side of connection be changed as specified by parameter in command sequence or returned to default value (DNR, WNR, or WNS)	cmd VTD,value cmd - DOR, DNR, WLR, WNR, WLS, or WNS (DOS and DNS are not supported by CCU/TAC TMPs) value - vertical tab disposition indicator: 0-250 = insert specified number of nulls following character (for time delays) 252 = replace vertical tab with end-of-line function 253 = simulate vertical tab function 254 = discard character 255 = deliver character as received

Page 4 of 4.

4.3.6 Mode Command

This command allows the user to specify the data packaging mode for his data going toward the network. The user would indicate that THP should enter record or stream mode (see Paragraph 3.3). The command will cause THP to send a new Set Mode record to the remote THP, informing that THP of the change. The user may not request a mode change for data coming toward him from the network.

4.3.7 Move Command

The move command allows the user to move his active connection to another user on the same MCCU and is available to MCCU users only. The feature is useful for a host which requires that one channel between an MCCU and host be a "logon" channel. Once the logon verification is complete, the host logon process can request that the connection be moved to another user on the same MCCU. The requester, of course, must have an active connection. In addition, the requester is responsible for ensuring that the new user and the current remote user have compatible characteristics and cross-connection matrix types. The new user, as specified in command parameters, must not have an established connection (opening or active), or be in a line disconnected state, i.e., must be closed, closing or listening. In addition, the user must meet the normal TCP requirements for the connection (security, precedence, TCC). If these conditions are met, the requester and new user will be informed of the move and THP will "map" the connection to the new user. The original user (requester of move) will then be available for additional connections. If the user is not valid according to TCP or if the specified port ID is not available, the user requesting the move will be informed and his connection continued as normal. The use of this command should be restricted to the logon channel of an MCCU. The automatic binary mode feature (see Paragraphs 2.2.1 and 5.3.1) will not be available for users that are allowed to enter a move command.

4.3.8 RCTE Command

The RCTE command is valid only if the RCTE option has been negotiated (see Paragraph 5.3.4). The connection, of course, must be active. The nature of the RCTE option, i.e., CCU user controls echo feature of TAC Mode IIA terminals and packet release of TAC Mode I, IB, IIA, and VI terminal users, makes the RCTE command available to CCU users

only. The command has two parameters, at least one of which must be specified for any one RCTE command sequence. The first parameter controls the hardware echo for the remote Mode IIA terminal (ON or OFF). The second controls the packet release for the remote terminal (one of standard packet release methods). TMP will build the appropriate RCTE control record and send it immediately to the remote TMP. That is, the RCTE commands act as packet release mechanisms for the requester, his letter will be sent as soon as the RCTE control record has been entered.

4.3.9 Time Command

The time command allows the user to manipulate, to some extent, the connection inactivity timer (see Paragraph 2.3.1) for his active connection. There will be a system generated value, on the order of 5 minutes, assigned to each user. This value can be changed at system generation time, if desired. If a zero value is assigned, there will be no connection inactivity timer. The time command will allow the user to activate and deactivate this timer during his connection. When the timer is inactive there may be long periods in which no data flows in either direction on the connection. Normally this idle condition would cause termination of the connection. It should be noted that the connection inactivity timer is separate from the other timers discussed in Paragraph 2.3.1. The user has no control over the other timers.

4.3.10 Echo and Duplex Commands

These commands, available to TAC Mode IIA terminal users only, allow the user to control the local hardware echo feature for his terminal. The echo command (see Table 4) requires an argument, ON or OFF. The hardware echo feature will be turned on or off, as directed. The full duplex command is the equivalent of an "echo on" command sequence. The half duplex command is the equivalent of the "echo off" command sequence. These commands are not available if the RCTE option (see Paragraph 5.3.4) has been negotiated for the user. Once the RCTE option has been negotiated, the user has forfeited control of his echo and packet release mechanism.

4.4 DATA TRANSMISSION CONTROL CHARACTERS

Data transmission control characters are special characters which are recognized by THP under certain conditions while scanning user-to-network data. Character values may be unique for each user and are defined at system generation time and changeable by the set command (see Paragraph 4.3.2). These characters are recognized and acted upon by THP (1) to perform local processing, (2) to convert the character to an NVT model value, ensuring recognition by remote user, or (3) to perform some action perhaps requiring THP-THP or THP-TCP communication. The data transmission control characters, as summarized in Table 7, are recognized and acted upon under the following conditions:

1. in NVT data scanning mode when detected in user data stream and not preceded by a prefix character (PC)

2. in binary or transparent data scanning mode when detected in data stream following a PC (the go-ahead character is an exception and is not effected by transparent mode)

3. a PC is expected to precede a THP command or data transmission control character; if the character following a PC is a second PC it will be transmitted as data in all cases

4. the first PC is always discarded.

TABLE 7. TMP DATA TRANSMISSION CONTROL CHARACTERS

CHARACTER NAME	FUNCTION DESCRIPTION	PARAGRAPH REFERENCE
prefix character (PC)	causes character(s) which follow it to be given special attention; effectively, TMP enters command mode	4.4.1
send now (SN) character	causes accumulated user text to be sent to network, acting as a packet release; character itself is not sent across network	4.4.2
are-you-there? (AYT) character	causes AYT record to be sent across network which will result in character being sent to remote user, presumably causing user status to be returned	4.4.3
erase character (EC) or erase line (EL) characters	causes EC or EL record to be sent across network which will result in appropriate editing character being sent to remote user	4.4.4
XASCII shift-out (SO) character	causes user data to be packaged in XASCII records with no special processing until an XASCII shift-in (SI) character is detected in user data stream; character is not recognized if XASCII option has not been negotiated	4.4.5
XASCII shift-in (SI) character	causes XASCII mode to be terminated and normal NVT mode to resume; character is not recognized if not currently in XASCII mode	4.4.5
go-ahead (GA) character	causes GA record to be sent across network which will result in character being sent to remote user indicating sender is ready to receive data; character is recognized only if go-ahead option has been negotiated	4.4.6
interrupt function 1/11 (IF1/11)	causes signal to be sent in-band to remote user	4.4.7
interrupt function 2/12 (IF2/12)	causes flush of data in network access areas (CCU/TAC) between this user and remote user	4.4.7
interrupt function 3/13 (IF3/13)	causes signal to be sent in-band to remote user after flush of data in network access areas (CCU/TAC) between this user and remote user	4.4.7
interrupt function 4/14 (IF4/14)	causes signal to be sent in-band to remote user and causes flush of data in network access areas (CCU/TAC) between remote user and this user (reverse flush)	4.4.7

Page 1 of 2.

TABLE 7, TMP DATA TRANSMISSION CONTROL CHARACTERS (CONT'D)

CHARACTER NAME	FUNCTION DESCRIPTION	PARAGRAPH REFERENCE
interrupt function 5/15 (IF5/15)	causes flush of data in network access areas (CCU/TAC) between remote user and this user (reverse flush); request is sent to remote TMP in-band	4.4.7
interrupt function 6/16 (IF6/16)	causes signal to be sent out-of-band to remote user	4.4.7
interrupt function 7/17 (IF7/17)	causes signal to be sent out-of-band to remote user and causes flush of data in network access areas (CCU/TAC) between remote user and this user	4.4.7
interrupt function 8/18 (IF8/18)	causes flush of data in network access areas (CCU/TAC) between remote user and this user; request sent to remote TMP out-of-band	4.4.7
interrupt function 9/19 (IF9/19)	causes one signal to be sent out-of-band to remote user and one signal to be sent in-band to remote user (at point of entry)	4.4.7
interrupt function 10 (IF10)	causes signal to be sent via TCP out-of-band control command to remote user; there is no network accountability	4.4.7

Page 2 of 2.

4.4.1 Prefix Character (PC)

The TMP prefix character (PC) is a very special character in that it allows the user to instruct TMP to enter command mode, essentially, to treat the next character(s) specially. The prefix character is never converted to an NVT value; it is either sent as data (second consecutive PC) or discarded upon entering command mode. The following rules apply to the prefix character:

1. PC must precede each (or the first in a series of) connection control and NVT control commands for them to be acted upon; PC is considered part of the command sequence

2. PC character value may not be changed (set command) to a value equal to any other data transmission control character, nor may any other character value be changed to that of the current PC

3. in NVT mode PC must not precede data transmission control characters (IF1 thru IF19, AYT, EC, EL, GA, SN, PC) for them to be acted upon; these characters will be sent as data if preceded by PC in NVT mode

4. in binary or transparent mode PC must precede data transmission control characters (IF1 thru IF19, AYT, EC, EL, GA, SN) for them to be acted upon; these will be sent as data if not preceded by prefix in these modes; GA will not be effected by transparent mode, i.e., NVT rules hold true for the GA character

5. PC and, therefore, TMP commands will not be recognized if user-to-network scanning mode is XASCII; only SI character is recognized in XASCII mode

6. second consecutive PC will always be sent as data

4.4.2 Send Now (SN) Character

The send now (SN) character, as specified in the user profile and when recognized by TMP, will act as a packet release mechanism. Upon detection of the SN character, TMP will complete the letter containing accumulated user data and send it to the network via TCP (Send event). The SN character is not sent across the network unless it is being treated as data. There are four such cases: in binary or transparent mode when SN is not preceded by a PC; in NVT

mode when SN is preceded by a PC; and in XASCII mode, (Also see Paragraphs 3.2 and 3.6).

4.4.3 Are-You-There? (AYT) Character

The are-you-there? (AYT) character, as specified in the user profile and when recognized by TMP, will act as a packet release mechanism. It is assumed that the function this invokes is important and, therefore, its delivery should be immediate. The AYT character is converted to NVT record, as appropriate for current data scanning mode, but does not cause any other processing in either local or remote TMP. The appropriate local character will be sent to the remote user by the remote TMP upon detection of the AYT record.

4.4.4 Erase Line (EL) and Erase Character (EC) Characters

The EC and EL characters, as specified by the user profile, will be recognized by TMP and converted to NVT records, as appropriate for the current data scanning mode. Corresponding characters will be passed to the remote user, as required by that user profile. No editing or other than NVT conversion processing will be accomplished by either local or remote TMP.

4.4.5 XASCII Shift-out (SO) and Shift-in (SI) Characters

The SO and SI characters, as specified in the user profile, will be recognized by TMP only if XASCII mode has been agreed upon via option negotiation (see Paragraph 5.3.3). If so data between the SO and SI characters will be packaged in XASCII records to be sent to the destination. The SO and SI characters will not be sent, however; the remote TMP will insert appropriate characters, for entering and exiting XASCII mode, based on the remote user's profile.

4.4.6 Go-Ahead (GA) Character

The GA character will be recognized, as specified in the user profile, if the go-ahead option has been agreed upon via option negotiation (see Paragraph 5.3.5). TMP will send a Go-Ahead record to the remote TMP, which will, in turn, send the go-ahead signal to his user. The GA acts as a packet release mechanism for the local user (it will be sent immediately along with any accumulated user-to-network

data). A more extensive discussion of the use of the go-ahead is found in Paragraph 5.3.5.

4.4.7 Interrupt Function (IF) Characters

There are ten separate interrupt functions (IF1 thru IF10). Nine of these (IF1 thru IF9) are duplicated in IF11 thru IF19, respectively, providing the user with two characters each for each of those functions. The tenth function (IF10) maps one character into a function which sends one character to the remote user. All interrupt functions act as packet release mechanisms for accumulated data. These ten functions will supply virtually every combination of interrupt/flush that would be required for normal user-to-user processing. Before defining the ten functions, some basic terms must be explained.

1. source - the side of the connection which initiates the interrupt function

2. destination - the side of the connection which did not initiate the interrupt function but is the recipient of a "signal-to-user" (see number 5, below)

3. in-band - processing occurs in-sequence, i.e., signal is passed or flush occurs following all data that was entered ahead of it; no attempt is made to process function ahead of data

4. out-of-band - there are two methods for handling functions out-of-band

   1. for IF6/16 thru IF9/19 - processing occurs out-of-sequence, i.e., character is converted to a TMP-TMP control record and passed or flush request (event to TCP) occurs ahead of data that was entered prior to the function request; this is done in TMP and destination HSI/TH only, i.e., source TMP will put the request for the function (Send event carrying a TMP letter containing any accumulated data and the interrupt function record) at the head of TCP's input queue; TCP will process the event normally; upon recognition of request, destination TMP will process immediately, putting request to send local character/signal (To User event) at the head of the HSI or TH input queue, as required by the function

   2. for IF10 - processing is out-of-sequence not only for TMP and HSI/TH but for

source/destination TCP as well; upon recognition of the character, source THP will queue the request for out-of-band interrupt to head of TCP input queue (Interrupt event); source TCP will pass request to destination TCP via the control information in a TCP header (this segment will not contain data and will not occupy sequence number space, i.e., no acknowledgement will be required for the segment), destination TCP will pass the request for an out-of-band interrupt to the head of THP queue (Interrupt Return event), THP will queue a request to send the appropriate character/signal to the head of HSI or TM input queue (To User event)

5. signal-to-user - function will involve sending appropriate character/signal to the destination user by destination THP when the function request is recognized (THP-THP control record or TCP event)

6. flush - in direction of flush (source to destination or destination to source), source THP and TCP discard all data currently in the source CCU or TAC; destination TCP discards data received from destination with sequence number lower than sequence number of the interrupt command segment; destination THP discards data currently awaiting processing upon notification by destination TCP (Interrupt Return event) and continues to discard records received until the THP data mark record is detected

7. data mark record - THP control record used to indicate when normal processing for a connection may resume, e.g., when a flush is complete

Currently defined interrupt functions are shown in Tables 7 and 8 and are summarized here. Detailed processing requirements may be determined based on the above definitions.

1. IF1/I1 - causes an IF1/I1 record to be added to the current to-network letter buffer and the letter to be sent immediately to the network (Send event to TCP); remote THP, upon detection of the IF1/I1 record sends an IF1/I1 character (or causes hardware signal to be sent, if appropriate) to the destination user

2. IF2/I2 - causes source THP to send a request to source TCP (Interrupt event for a flush function; source THP will discard any data awaiting transmis-

sion to the network; source TCP will cause the source to destination flush as described above

3. IF3/13 - causes source THP to discard current to-network letter buffer and build an IF3/13 record in a new to-network letter buffer; before sending this letter to the destination THP, source THP requests that source TCP perform a source to destination flush; destination THP will send the local IF3/13 character to the destination user upon detection of the IF3/13 record

4. IF4/14 - causes an IF4/14 record to be added to the current to-network letter buffer and the letter to be sent immediately to the network (Send event to TCP); destination THP, upon recognition of the IF4/14 record, will send the local IF4/14 character to the destination user and request that destination TCP perform a destination to source flush (Interrupt event)

5. IF5/15 - causes an IF5/15 record to be added to the current to-network letter buffer and the letter to be sent immediately to the network (Send event to TCP); destination THP, upon recognition of the IF5/15 record, will request that destination TCP perform a destination to source flush (Interrupt event)

6. IF6/16 - causes an IF6/16 record to be added to the current to-network letter buffer and the letter to be sent immediately to the network; this is the same as IF1/11 except that processing is handled "out-of-band" as described above

7. IF7/17 - causes an IF7/17 record to be added to the current to-network letter buffer and the letter to be sent immediately to the network; this is the same as IF4/14 except that processing is handled "out-of-band" as described above

8. IF8/18 - causes an IF8/18 record to be added to the current to-network letter buffer and the letter to be sent immediately to the network; this is the same as IF5/15 except that processing is handled "out-of-band" as described above

9. IF9/19 - causes an IF9/19 record to be sent in a separate to-network letter (processed out-of-band as described above) and another IF9/19 record to be added to the current to-network letter buffer and that buffer processed in-band as described above; this function, effectively, combines IF6/16 and

IF1/11 into one function

10. IF13 - causes source TMP to request that source TCP perform the out-of-band interrupt function with no network accountability, as described above

TABLE 8, INTERRUPT FUNCTION MATRIX

FUNCTION / INTERRUPT	IN BAND	SIGNAL TO USER	FLUSH S TO D	FLUSH D TO S	DATA MARK	OUT OF BAND
IF1/11	YES	YES	NO	NO	NO	NO
IF2/12	YES	NO	YES	NO	YES	NO
IF3/13	YES	YES	YES	NO	YES	NO
IF4/14	YES	YES	NO	YES	YES	NO
IF5/15	YES	NO	NO	YES	YES	NO
IF6/16	NO	YES	NO	NO	NO	YES
IF7/17	NO	YES	NO	YES	YES	YES
IF8/18	NO	NO	NO	YES	YES	YES
IF9/19	YES	YES	NO	NO	NO	YES
IF10	NO	YES	NO	NO	NO	YES

DEFINITIONS

in-band - processing occurs in-sequence, i.e., signal is passed or flush occurs following all data that was entered ahead of it; no attempt is made to handle function first

out-of-band - (1) for IF6/F16 through IF9/19 - processing occurs out-of-sequence, i.e., signal is passed or flush occurs before data that was entered ahead of it; this is done in TNP and destination HSI/TH only, i.e., source TNP will put request for function at head of TCP queue, TCP will process normally, upon recognition of request destination TNP will process first (for that connection), putting at head of TH or HSI queue, if called for in function; (2) for F10 - processing is out-of-band not only for TNP but for source/destination TCP as well; signal passed via TCP header control; TCP sequence space not used, therefore, no acknowledgement for segment; always queued to head of input queues

signal-to-user - local character or hardware signal, if desired and possible, is passed to destination host or terminal by remote TNP when request is recognized

flush - in direction of flush (Source to Destination or Destination to Source)—local TNP and TCP throw away all data currently in box; remote TCP throws away all data received with sequence number lower than sequence number of interrupt command segment; remote TNP throws away all data currently awaiting processing when notified of flush request (by remote TCP) and continues to throw away until TNP data mark record is received

data mark - TNP data mark record, used to indicate when normal processing for connection may resume, e.g., when flush is complete.

SECTION 5, OPTION NEGOTIATION

5.1 GENERAL INFORMATION

Option negotiation is the facility by which TMPs can agree to vary the normal NVT model processing for data flowing on a connection. The feature is provided so that a TMP with sufficient knowledge of his user or his user's data can request an option which will eliminate unnecessary processing for either TMP (sender or receiver of data). This knowledge may be acquired based on his user's profile or through option commands entered by the user (see Paragraph 4.3.5).

There are fourteen options defined for AUTODIN II. There is no requirement upon specific TMPs to implement all or any of these options. The option negotiation mechanism allows rejection as well as acceptance of a request to start any option. These and other rules and suggestions for implementation of option negotiation are discussed in Paragraph 5.2.

5.2 PROCEDURES

Option negotiation will occur in two situations for an active connection. First, it will occur automatically when a connection is first established. As discussed in Paragraph 2.1, characteristics option records are exchanged before any data. Other options may be negotiated at that time depending upon the outcome of the characteristics option negotiation (see Paragraph 5.3.1). These negotiations are optional, of course, and will not be done by CCU/TAC THPs. Second, when a user enters an option command (see Paragraph 4.3.5), negotiation for that option will be performed, as required, by THP. In either situation the same rules and option record formats apply.

Options are negotiated for only one side of a connection at a time. That is, there are two possible data paths for any connection. There are two THPs processing these data paths (one at either end of the connection). For each THP there is a send data path (user-to-network data) and a receive data path (network-to-user data). Any option is negotiated for only one of these data paths per option request.

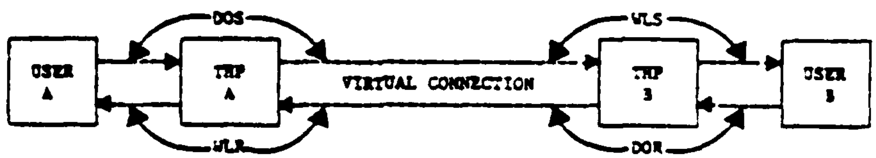
The following procedures must be followed by a THP which intends to interface with a CCU/TAC THP, and will be enforced, as indicated, by CCU/TAC THPs:

1. Option requests are represented by standard THP option records, the general format of which are defined in Appendix A. Specific formats for each option request are defined in Appendix B. Option requests have "operators" which are contained in the option record and correspond to user option commands (see Paragraph 4.3.5). These operators define which side of the connection is being discussed and which THP (sender or receiver of the data) will handle the option. Figure 1 illustrates this concept. The defined operators and the acronym and definition of each are:

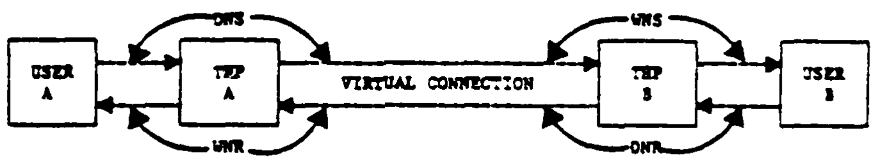
   1. do sender (DOS) - the requester desires to start an option on his send data path and he agrees to do the appropriate processing

   2. don't sender (DNS) - the requester desires to stop a previously started option on his send data path for which he had agreed to do the appropriate processing

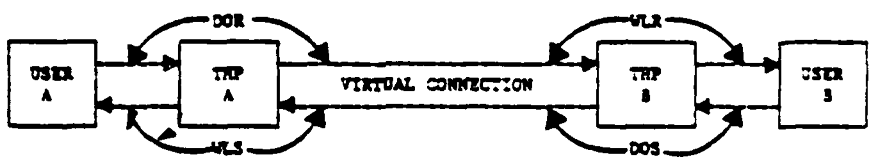
   3. do receiver (DOR) - the requester desires to start an option on his send data path and he agrees that the other THP (data receiver) may do the appropriate processing



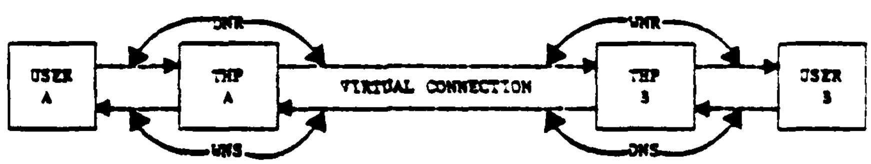
CASE ONE. Appropriate option commands/record types resulting in agreements for THP-A to start processing for an option for either side of a virtual connection.



CASE TWO. Appropriate option commands/record types resulting in agreements for THP-A to stop processing for an option that was previously negotiated for either side of the virtual connection.



CASE THREE. Appropriate option commands/record types resulting in agreements for THP-B to start processing for an option for either side of a virtual connection.



CASE FOUR. Appropriate option commands/record types resulting in agreements for THP-B to stop processing for an option that was previously negotiated for either side of the virtual connection.

Figure 1. Option Negotiation Request Paths

4. don't receiver (DNR) - the requester desires to stop a previously started option on his send data path for which he agreed that the other THP would do the appropriate processing

5. will sender (WLS) - the requester desires to start an option on his receive data path and he agrees that the other THP (data sender) will do the appropriate processing

6. won't sender (WNS) - the requester desires to stop a previously started option on his receive data path for which he had agreed that the other THP would do the appropriate processing

7. will receiver (WLR) - the requester desires to start an option on his receive data path and he agrees to do the appropriate processing

8. won't receiver (WNR) - the requester desires to stop a previously started option on his receive data path for which he had agreed to do the appropriate processing.

2. Each operator has a corresponding positive (agrees with request) and negative (refuses request) response. Only one of the indicated responses will resolve the pending option request, and thus stop the option timer (see procedure 5, below). It should be noted that negative responses to a request to stop an option are not permitted (see procedure 4, below)

request	positive	negative
DOS	WLS	WNS
DNS	WNS	not allowed
DOR	WLR	WNR
DNR	WNR	not allowed
WLS	DOS	DNS
WNS	DNS	not allowed
WLR	DOR	DNR
WNR	DNR	not allowed

3. Some option requests require parameters (see Appendix B). Parameters are not required for these if the request is to stop an option (DNS, DNR, WNS, or WNR).

4. A request to stop an option (DNS, DNR, WNS, or WNR) must be answered positively as these indicate a request to return to a known default condition. Refusal to return to the default will be treated as

a protocol error.

5. Valid option requests must always be answered, either with a positive or negative response. A timer will be used to ensure that a response is received within a reasonable period (on the order of 3 minutes). This timer is separate from the user's connection inactivity timer. If the timer expires, it will be considered a protocol error unless a flush has occurred on the connection during the option negotiation. If this is the case the currently pending option requests will be sent again with the retransmit indicator set (see Appendices A and B and procedure 6, below).

6. Requests should not be sent to announce the current state of an option, e.g., DNS, DNR, WNS, or WNR records, requesting that an option be terminated that is not currently being performed, or DCS, DCR, WLS, or WLR records, requesting that an option be started that is currently being performed. These duplicate requests will be treated as protocol errors when detected.

   Duplicate requests such as these would cause unterminating acknowledgement loops which would, essentially, hang up one side of the connection. It is assumed that a violation of this rule indicates that, either the TMP is not conforming to AUTODIN II-standard option negotiation procedures or a catastrophic situation exists in the remote access area (e.g., software failure). In either case, it is considered to be appropriate to close the connection.

   It is necessary to make one exception to this procedure. If the retransmission indicator (see Appendices A and B and procedure 5, above) is set in the option record, the request is handled regardless of the current state of the option. This exception is necessary because of the possibility of a flush of user data while option negotiation is in progress. If an interrupt function 2/12, 3/13, 4/14, 5/15, 7/17, or 8/18 is performed while option negotiation is pending, and the option timer subsequently expires, the option request will be resent. It is possible that either the request or the response option record was lost in the TCP flush.

7. Requests to start an option may always be rejected as this forces the known default for processing data. For CCU/TAC users an option may be set as nonnegotiable for a certain data path,

Corresponding requests would be rejected by THP.

8. Option requests which are rejected should not be repeated unless requested by local user, that is, THP should not resend the request arbitrarily.

9. A response, presumably negative, must be returned even if the option type is unknown, if the option request is to start an unknown option. If a response is not received the option timer will expire.

10. A request to stop an unknown option is considered a protocol error.

11. THP should buffer (hold) user-to-network data if option being negotiated for that data path. This will provide a clear marker for the beginning of option processing; when the option is agreed upon, data will be in new format

12. CCU/TAC THPs will not send DOS option records. This implementation decision was made because the onus for processing data is on the data receiver; i.e., the receiving THP must be able to convert NVT model format to the format required by his user's characteristics. Therefore, the bulk of option processing is contained in the "receive side" of CCU/TAC THPs and is not duplicated in the "send side." The binary mode, XASCII, RCTE, and go-ahead options are exceptions as both THPs are involved, however, these require receiver-oriented operators (see procedure 13, below).

13. Option types which require that both THPs perform some processing, such as, characteristics, binary mode, RCTE, XASCII, and go-ahead, require the use of receiver-oriented operators (DOR, DNR, WLR, WNR) for user option commands and THP option records.

14. When the binary, RCTE, go-ahead, or XASCII options are negotiated (started or stopped) the non-requesting user will be informed about the negotiations. The requesting user will always be informed of the results of the option negotiation. (See Paragraph 4.3.5).

For each option certain alternatives are available and a known default is defined. These are summarized in Table 9 and subsequent paragraphs. The characteristics option is not represented in Table 9 as it is a unique case, one which may not be invoked by the user nor negotiated other than during open processing (see Paragraph 2.2).

5.3 VALID OPTIONS

5.3.1 Characteristics Option

The characteristics option is requested during open processing (see Paragraph 2.2) and is unique in that it carries with it the cross-connection compatibility check, which is an AUTODIN II requirement. If the users are not compatible according to the cross-connection matrix, the connection will be closed immediately.

The characteristics check involved in this option is performed to determine if the two users have matching characteristics, i.e., if data being sent by one user is in the required format for the data receiver. If this is the case, the positive acknowledgement will put that side of the connection in binary mode. The assumption is that if the users characteristics complement one another, there is no requirement for NVT conversion. Binary mode is automatically entered on behalf of the user when the positive response is received for the characteristics option. Both users will be informed that the particular data path is in binary mode. All rules of binary mode apply, including exiting binary mode via user option command. If the characteristics do not match, the users will be informed that the particular data path is in NVT mode.

Because this option affects the processing of data, is part of open processing, and most importantly, involves the cross-connection matrix validation, data will be accumulated from the user, as buffering permits, but not sent to the network until negotiation is complete. Each THP sends characteristics of his user's receive side (WLR), i.e., how the data should look coming toward his user. The option requires no sender or receiver qualifier as both THPs are involved, therefore, the default "will receiver" option record is used.

5.3.2 Binary Mode Option

Binary mode, as discussed in Paragraphs 3.1, 3.4, and 3.5, can be entered automatically as a result of the characteristics option (see Paragraph 5.3.1) or upon agreement by THPs at the request of the user. Because both THPs must be involved in binary mode processing and the option does not require sender or receiver qualifiers, receiver operators and commands are used, as appropriate. Data going toward the network, that is, if in the direction of the data path being discussed, will be held until an agreement or rejection is received. This procedure leaves no doubt as to where one

mode ends and another begins. As soon as the option is agreed to, data received/sent for that side of the connection will be in binary mode. CCU/TAC TMPs will accumulate user data (8-bit characters) in Data records regardless of the packaging mode (record or stream).

5.3.3 XASCII Mode Option

XASCII mode, as discussed in Paragraphs 3.1, 3.4, and 3.5, will be negotiated by TMPs at the request of the user. The ability to enter XASCII mode must be agreed upon before XASCII data may be sent, but once the agreement is made and unless it is later revoked, XASCII mode may be entered any time (except while in binary mode) for the duration of the connection. When in XASCII mode, data is passed with no processing other than building XASCII records. The parity bit will not be stripped from each octet received from the user as is done in NVT mode processing. If XASCII mode is agreed upon, TMP will scan user-to-network data for XASCII shift-out (SO) character. If detected, XASCII scanning mode will be entered and no character other than XASCII shift-in (SI) will be recognized. That is, no TMP commands, including data transmission control characters, will be acted upon.

The XASCII option is similar to binary mode and characteristics options in that the option requires no sender or receiver qualifier. The receiver-oriented operators will be used.

5.3.4 RCTE Option

The RCTE option provides the host process (via CCU) control over two aspects of a TAC terminal to which he is connected. If the terminal is a Mode IIA type terminal, the host process can control the hardware echo feature (see Paragraph 4.3.12). For modes IIA, I, IB, and VI the host process can control the packet release to be used in releasing terminal data to the network. This control is conveyed in RCTE control records which are sent as a result of user RCTE commands in the CCU (see Paragraph 4.3.8). However, the RCTE option must have been negotiated previously by TMPs as a result of a user option command. It should be noted that if a terminal user agrees to negotiate to start the RCTE option (option is marked as negotiable in his user profile), the terminal user should understand that he has forfeited the control of his packet release mechanism and echo feature (see Paragraphs 4.3.3 and 4.3.10).

The receiver-oriented operators are used for the RCTE option. WLR and WNR are appropriate commands/operators for a CCU; DOR and DNR are appropriate for a TAC. This is because it is the TAC send side of the connection being negotiated, not the direction of the RCTE commands (see Paragraph 4.3.3).

5.3.5 Go-Ahead Option

The go-ahead option provides a means by which go-ahead signals may be exchanged on a connection. A go-ahead (GA) signal could serve two purposes: (1) tell receiver of GA that sender of GA is ready to receive data or (2) tell receiver of GA to send more data. Functionally, these two seem the same. However, for the first case, the sender of the GA can not receive data (i.e., his terminal configuration is essentially half duplex) until the GA is sent by him. In the second case, the receiver of the GA will not send until "prompted" for data by the GA character. In case one, the local TNP may be able to facilitate the function by sending data to his user when a GA is detected coming from his user. The remote user or TNP need not necessarily be informed of the GA signal, e.g., if he has no requirement for it.

The go-ahead compatibility will be verified during characteristics option processing (see Paragraph 5.3.1) during the opening of a connection. If an incompatibility is detected, i.e., if one user requires go-ahead signals, and the other user will not send the GA, the connection will be closed immediately. Both users will be informed. The connection would not be able to exist under those circumstances; a stand-off would eventually ensue in which one user was waiting to be told to send and the other waiting to receive.

The receiver-oriented operators are used for the go-ahead options. The appropriate operators WLR/WNR or DOR/DNR are chosen based on which side of the connection the go-ahead signal will be sent.

5.3.6 Line Width Option

The line width option allows TNPs to agree on (1) the number of characters per line to be used in formatting data on the data path being referenced and (2) which TNP will handle the formatting. The line width requested should be within a reasonable limit for the data receiver (e.g., within hardware limitations), but this must be determined by the TNP serving the data receiver. There seems to be no other restrictions since the sender of the data may know more

about appropriate format for the data and, therefore, would be qualified to specify the line width. Of course, local user option or set commands (see Paragraphs 4.3.2 and 4.3.5) changing this characteristic will override network option requests or agreements. The option would have to be renegotiated.

5.3.7 Page Size Option

The page size option allows TMPs to agree on (1) the number of print lines per page to be used in formatting data on the data flow being referenced and (2) which TMP will handle the formatting. The page size should be within a reasonable limit for the data receiver (e.g., within hardware limitations), but this will be determined by the TMP serving the data receiver.

5.3.8 Tab Stop Options

There are two tab stop options, vertical and horizontal, which allow TMPs to agree upon settings for tabs. Actual disposition, how tabs will be processed, is negotiated in the horizontal or vertical tab disposition options (see Paragraph 5.3.9). There are two alternatives to setting tabs: (1) every Nth column/line, beginning at N, or (2) maximum of six columns/lines specified. Settings should be within the bounds of the user's characteristics (hardware paper width and length), but this will be determined by the TMP serving the data receiver.

5.3.9 NVT Character Disposition Options

There are five options which allow TMPs to agree on (1) variations to NVT default processing for certain characters (CR, LF, FF, VT, HT) and (2) which TMP will handle disposition. These are: carriage return, linefeed, formfeed, vertical tab, and horizontal tab disposition options. For any disposition there are certain variations available. These are shown in Table 9, but are defined here for clarification:

1. as is - pass character with no additional delays or simulations: all characters (CR, LF, FF, VT, HT)

2. discard - do not pass character when detected, in user data stream (if sending TMP) or in network data stream, NVT record (if receiving TMP); all

characters

3. simulate - replace character with defined simulation characters as follows:

   1. CR - no simulation available

   2. LF - insert new line function and enough spaces to position one line below current vertical position

   3. FF - perform new line function and enough LF functions to position to top of next page (leftmost column on first line of page)

   4. VT - perform new line function and enough LF functions to position to next tab line; if currently positioned past last tab, perform FF function

   5. HT - insert enough ASCII spaces to position to next tab column; if currently beyond last tab, perform new line function

4. replace with CR/LF - replace character with CR/LF or new line function; FF and VT only

5. add delays - insert, following character, specified number of nulls for timing delays; all characters

TABLE 9, TMP OPTION MATRIX

OPTION	KNOWN DEFAULT CONDITION	ASSIGN VALUES (1-250)	DELIVER AS IS (255)	DISCARD CHARACTER (254)	SIMULATE CHARACTER (253)	REPLACE WITH NEWLINE (252)	ADD DELAY (0-250)	INCREMENTAL VALUE (255)
BINARY MODE	NVT SCANNING MODE	NA	NA	NA	NA	NA	NA	NA
RCTE	COMMANDS NOT VALID	NA	NA	NA	NA	NA	NA	NA
GO-AHEAD	WILL NOT REACT TO GO-AHEAD	NA	NA	NA	NA	NA	NA	NA
XASCII MODE	WILL NOT REACT TO XASCII SHIFT-OUT	NA	NA	NA	NA	NA	NA	NA
LINE WIDTH	MOST RECENT SET BY LOCAL USER	YES	NA	NA	NA	NA	NA	NA
PAGE SIZE	MOST RECENT SET BY LOCAL USER	YES	NA	NA	NA	NA	NA	NA
HORIZONTAL TAB STOPS	MOST RECENT SET BY LOCAL USER	YES, UP TO SIX TABS	NA	NA	NA	NA	NA	YES
VERTICAL TAB STOPS	MOST RECENT SET BY LOCAL USER	YES, UP TO SIX TABS	NA	NA	NA	NA	NA	YES
CARRIAGE RETURN (CR) DISPOSITION	CONVERTED TO NVT CR RECORD WITH NO DELAYS	NA	YES	YES	NA	NA	YES	NA
LINEFEED (LF) DISPOSITION	CONVERTED TO NVT LF RECORD WITH NO DELAYS	NA	YES	YES	IF NO CR BEFORE IT REPLACE W/CR AND SPACE OVER	NA	YES	NA

Page 1 of 2.

LEGEND: NA - not a negotiable item for this option
YES- will negotiate item for this option

TABLE 9, TMP OPTION MATRIX (CONT'D)

OPTION	KNOWN DEFAULT CONDITION	ASSIGN VALUES (1-250)	DELIVER AS IS (255)	DISCARD CHARACTER (254)	SIMULATE CHARACTER (253)	REPLACE WITH NEWLINE (252)	ADD DELAY (0-250)	INCREMENTAL VALUE (255)
FORMFEED (FF) DISPOSITION	CONVERT TO NVT FF RECORD WITH NO DELAYS	NA	YES	YES	REPLACE W/CR AND ENOUGH LF TO GET TO TOP OF NEXT PAGE	YES	YES	NA
HORIZONTAL TAB (HT) DISPOSITION	CONVERT TO NVT HT RECORD WITH NO DELAYS	NA	YES	YES	INSERT SPACES TO NEXT TAB POSITION; IF BEYOND LAST TAB POSITION INSERT CR AND LF	NA	YES	NA
VERTICAL TAB (VT) DISPOSITION	CONVERT TO NVT VT RECORD WITH NO DELAYS	NA	YES	YES	INSERT CR AND ENOUGH LF TO GET TO NEXT TAB POSITION; IF BEYOND LAST TAB POSITION INSERT FF	YES	YES	NA

Page 2 of 2.

LEGEND: NA - not a negotiable item for this option
YES- will negotiate item for this option

INDEX

APPENDIX A

THP RECORDS

This appendix describes the THP-THP control record format to be used by a THP desiring to interface with CCJ/TAC THPs on the AUTODIN II network. There is no variation to these formats as described herein.

Included in the description of each record is the name, accepted acronym, record type value (decimal), a brief description of the record and its purpose, and the format for the record.

The format is shown in standard DEC PDP-11 memory image format, with the least significant byte on the right (even number) and the most significant byte on the left (odd number) of a 16-bit word. The bits are labeled 0 to 15 (least significant to most significant) right to left in the word. The direction of transmission is from the least significant byte/least significant bit (0) to the most significant byte/most significant bit (15) for each 16-bit word.

The memory image for each record shown in this appendix is described by an accompanying narrative. The size (byte, word, etc.) of each unit within the structure may cause an "extra" byte at the end of the record. This extra byte is not used and will not be defined in the narrative. It will be represented by a series of dashes (- - -) in the memory image. The extra byte will not be counted in the length of the structure and in fact is not sent as part of the record.

The records are listed in alphabetical order by the first word in the record name, as referenced in the text of this document.

RECORD NAME: Are-You-There?
RECORD TYPE ACRONYM: AYT
RECORD TYPE VALUE: 219
DESCRIPTION: The AYT record is built in the to-network buffer when the local AYT character is recognized in the user's data stream. The destination THP will convert the record to the destination user's AYT character or signal and send it to that user. The purpose is to allow the user to have some method of determining if the remote user is still actively participating on the connection. The AYT character is represented by an AYT record whether the connection is in record or stream mode.

RECORD FORMAT:

```
        15 . . . . 7        0
        ---------------------
        |        |         |
     1  |  AYT   |   RM    |  0
        |        |         |
        ---------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of record.

BYTE 1: AYT record type value (219).

RECORD NAME: Backspace
RECORD TYPE ACRONYM: BS
RECORD TYPE VALUE: 215
DESCRIPTION: The BS record is build in the to-network buffer when the local BS character is recognized in the user's data stream. The destination THP will convert the record to the destination user's BS character and send it to that user. This is an NVT character which affects the NVT printer and as such is included in those characters that are not sent in record form when the connection is in stream mode. The NVT value (record type value) is sent in place of the local user's BS character value.

RECORD FORMAT:

```
        15  . . .  7         0
        --------------------
        !        !         !
    1   !   BS   !   RM    !   0
        !  . . . !         !
        --------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: BS record type value (215).

RECORD NAME: Bell
RECORD TYPE ACRONYM: BEL
RECORD TYPE VALUE: 214
DESCRIPTION: The BEL record is added to the to-network buffer when the local BEL character is recognized in the user's data stream. The destination THP will convert the record to the destination user's BEL character and send it to that user. The BEL character is an NVT character which affects the NVT printer and is not sent in record form when the connection is in stream mode. The NVT value (record type value) is sent in place of the local user's BEL character.

RECORD FORMAT:

```
         15         7         0
         --------------------
         !        !         !
    1    !   BEL  !   RM    !   0
         !        !         !
         --------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: BEL record type value (214).

RECORD NAME: Carriage Return Character
RECORD TYPE ACRONYM: CR
RECORD TYPE VALUE: 209
DESCRIPTION: The CR record is added to the to-network buffer when the local CR character is recognized in the user's data stream. The destination THP will convert the record to the destination user's CR character and send it to that user. The CR character is an NVT character which affects the NVT printer and is not sent in record form when the connection is in stream mode. The NVT value (record type value) is sent in place of the local user's CR character.

RECORD FORMAT:

```
         15        7        0
         -------------------
         |        |        |
      1  |   CR   |   RM   |  0
         |        |        |
         -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: CR record type value (209).

RECORD NAME: Data
RECORD TYPE ACRONYM: DATA
RECORD TYPE VALUE: 221
DESCRIPTION: The DATA record contains the text from the user's input data stream. The text is directly transferrable to the tomuser buffer with no additional character scanning required. All NVT-related characters have been converted, as required; therefore, it will not be necessary to scan the DATA record for these characters.

It should be noted that DATA records are used during NVT mode for user text and during binary mode for "binary" data. Although user text is not ordinarily sent in records when the connection is in stream mode, binary data is always sent in DATA records by CCU/TAC THPs.

RECORD FORMAT:

```
      15        7        0
      -------------------
      |        |        |
  1   |  DATA  |   RM   |   0
      |        |        |
      -------------------
      |                 |
  3   |     length      |   2
      |                 |
      -------------------
      |        |        |
  5   |  text  |  text  |   4
  .   |        |        |   .
  .   -------------------   .
  .   |        |        |   .
  n   |  text  |  text  |  n-1
      |        |        |
      -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: DATA record type value (221).

BYTE 2: Low-order 8 bits of 16-bit length value. Length is a value representing number of bytes of text that are in the DATA record, i.e., length of DATA record beyond the length field.

BYTE 3: High-order 8 bits of 16-bit length value. Length is a value representing number of bytes of text that are in the DATA record, i.e., length of DATA record beyond the length field.

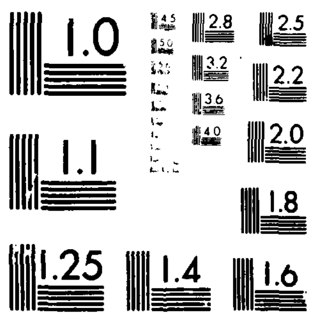
BYTES 4 thru "n": Text characters to be passed to the user. Characters are 7-bit ASCII with parity bit stripped by the sending THP except for binary mode. In binary mode the entire octet (8 bits), received from the user, is sent in the DATA record.

****
11-JUL-79 T0844939 **UNCLASSIFIED**
****

AD-A081 757 WESTERN UNION TELEGRAPH CO MCLEAN VA GOVERNMENT SYST--ETC F/6 17/2
COMPUTER PROGRAM DEVELOPMENT SPECIFICATION TERMINAL-TO-HOST PRO--ETC(U)
FEB 80 DCA200-C-637

UNCLASSIFIED SBIE-AD-E100 340 NL

2 OF 4
AD
A081757



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RECORD NAME: Data Mark
RECORD TYPE ACRONYM: DM
RECORD TYPE VALUE: 255
DESCRIPTION: The DM record is inserted in the to-network buffer to indicate the end of a flush function. When the destination THP receives the DM record a flush of network-to-user records, currently in progress, will be terminated. With this record there is a clear marker for the end of flush processing. The interrupt function record would have been received previously by the destination THP, informing destination THP to begin the flush.

RECORD FORMAT:

```
           15    .   .   7          0
           ---------------------------
           |            |            |
       1   |     DM     |     RM     |   0
           |        .   .|           |
           ---------------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: DM record type value (255).

RECORD NAME: Do Receiver Option Negotiation Record
RECORD TYPE ACRONYM: DOR
RECORD TYPE VALUE: 235
DESCRIPTION: The DOR option negotiation record is added to the to-network buffer when the local user enters an option negotiation command, or during open processing when the characteristics option is negotiated. The detailed description of each option type record is shown in Appendix B. This shows the general format for an DOR option negotiation record, without regard to option type.

The DOR option negotiation record indicates that the specified option should be started on the requester's send side. The requester agrees that the other TMP (data receiver) will handle processing for the option.

The DOR option negotiation record is also used for those option types which require that both TMPs do some processing for the option, such as, characteristics, binary, RCTE, XASCII, and go-ahead.

RECORD FORMAT:

```
        15       7        0
        -------------------
        !        !        !
   1    !  DOR   !   RM   !   0
        !        !        !
        -------------------
        !                 !
   3    !     length      !   2
        !                 !
        -------------------
        !        ! !      !
   5    ! param  !R! type !   4
        !        ! !      !
        -------------------
        !        !        !
   7    ! param  ! param  !   6
        !        !        !
        -------------------
        !        !        !
   9    ! param  ! param  !   8
        !        !        !
        -------------------
        !        !        !
  11    ! -----  ! param  !  10
        !        !        !
        -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: DOR record type value (235).

BYTE 2: Low-order 8 bits of 16-bit length field. Length value represents number of bytes in the record beyond the length field, i.e., first four bytes of the record are not counted in length value.

An option negotiation record will not be longer than 38 bytes total i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 3: High-order 8 bits of 16-bit length field. Because option records will be no longer than 38 bytes, this byte will always be zero for option records.

BYTE 4: Option type value/retransmission indicator. The high-order bit (R) is set if this record is a retransmission. The low-order 7 bits represent the value for one of the fourteen currently defined option types for AUTODIN II:

value	option type
0	characteristics compatibility
1	binary mode
2	remote control of transmission and echo (RCTE)
3	go-ahead
4	extended ASCII (XASCII)
5	line width
6	page size
7	horizontal tab settings
8	vertical tab settings
9	carriage return disposition
10	linefeed disposition
11	formfeed disposition
12	horizontal tab disposition
13	vertical tab disposition

BYTES 5 thru 10: Additional parameter bytes for some options, such as tab settings or line width/page size value.

RECORD NAME: Don't Receiver Option Negotiation Record
RECORD TYPE ACRONYM: DNR
RECORD TYPE VALUE: 236
DESCRIPTION: The DNR option negotiation record is added to the to-network buffer when the local user enters an option negotiation command, or during open processing when the characteristics option is negotiated. The detailed description of each option type record is shown in Appendix B. This shows the general format for an DNR option negotiation record, without regard to option type.

The DNR option negotiation record indicates that the specified option should be stopped on the requester's send side.

Values, expressed in the parameter field of the DOR record, are not required in the request to stop the option (DNR record).

RECORD FORMAT:

```
            15          7         0
            ---------------------
            |          |         |
       1    |   DNR    |   RM    |   0
            |          |         |
            ---------------------
            |          |         |
       3    |     length         |   2
            |          |         |
            ---------------------
            |          | |       |
       5    |  -----   |R| type  |   4
            |          | |       |
            ---------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: DNR record type value (236).

BYTE 2: Low-order 8 bits of 16-bit length field. Length value represents number of bytes in the record beyond the length field, i.e., first four bytes of the record are not counted in length value.

The DNR record will require no values, therefore, the length will always be 1.

BYTE 3: High-order 8 bits of 16-bit length field. This byte will be zero for all option records.

BYTE 4: Option type value/retransmission indicator. The high-order bit (R) is set if this record is a retransmission. The low-order 7 bits represent the value for one of the fourteen currently defined option types for AUTODIN II:

value	option type
0	characteristics compatibility
1	binary mode
2	remote control of transmission and echo (RCTE)
3	go-ahead
4	extended ASCII (XASCII)
5	line width
6	page size
7	horizontal tab settings
8	vertical tab settings
9	carriage return disposition
10	linefeed disposition
11	formfeed disposition
12	horizontal tab disposition
13	vertical tab disposition

RECORD NAME: Do Sender Option Negotiation Record
RECORD TYPE ACRONYM: DOS
RECORD TYPE VALUE: 233
DESCRIPTION: The DOS option negotiation record is added to the to-network buffer when the local user enters an option negotiation command, or during open processing when the characteristics option is negotiated. The detailed description of each option type record is shown in Appendix B. This shows the general format for an DOS option negotiation record, without regard to option type.

The DOS option negotiation record indicates that the specified option should be started on the requester's send side.

RECORD FORMAT:

```
          15 . . .  7       0
          --------------------
          |        |        |
       1  |  DOS   |   RM   |  0
          |        |        |
          --------------------
          |        |        |
       3  |     length      |  2
          |        |        |
          --------------------
          |        | |      |
       5  | param  |R| type |  4
          |        | |      |
          --------------------
          |        |        |
       7  | param  | param  |  6
          |        |        |
          --------------------
          |        |        |
       9  | param  | param  |  8
          |        |        |
          --------------------
          |        |        |
      11  | -----  | param  |  10
          |        |        |
          --------------------
```

BYTE 0: TMP standard record mark value (170), denoting beginning of TMP record.

BYTE 1: DOS record type value (233).

BYTE 2: Low-order 8 bits of 16-bit length field. Length value represents number of bytes in the record beyond the length field, i.e., first four bytes of the record are not counted in length value.

An option negotiation record will not be longer than 38 bytes total i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 3: High-order 8 bits of 16-bit length field. Because option records will be no longer than 38 bytes, this byte will always be zero for option records.

BYTE 4: Option type value/retransmission indicator. The high-order bit (R) is set if this record is a retransmission. The low-order 7 bits represent the value for one of the fourteen currently defined option types for AUTODIN II:

value	option type
0	characteristics compatibility
1	binary mode
2	remote control of transmission and echo (RCTE)
3	go-ahead
4	extended ASCII (XASCII)
5	line width
6	page size
7	horizontal tab settings
8	vertical tab settings
9	carriage return disposition
10	linefeed disposition
11	formfeed disposition
12	horizontal tab disposition
13	vertical tab disposition

BYTES 5 thru 10: Additional parameter bytes for some options, such as tab settings or line width/page size value.

RECORD NAME: Don't Sender Option Negotiation Record

RECORD TYPE ACRONYM: DNS

RECORD TYPE VALUE: 234

DESCRIPTION: The DNS option negotiation record is added to the to-network buffer when the local user enters an option negotiation command, or during open processing when the characteristics option is negotiated. The detailed description of each option type record is shown in Appendix B. This shows the general format for an DNS option negotiation record, without regard to option type.

The DNS option negotiation record indicates that the specified option should be stopped on the requester's send side.

RECORD FORMAT:

```
        15        7        0
        --------------------
        !        !         !
    1   !  DNS   !   RM    !   0
        !        !         !
        --------------------
        !        !         !
    3   !     length       !   2
        !        !         !
        --------------------
        !        ! !       !
    5   !  ----- !R! type  !   4
        !        ! !       !
        --------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: DNS record type value (234).

BYTE 2: Low-order 8 bits of 16-bit length field. Length value represents number of bytes in the record beyond the length field, i.e., first four bytes of the record are not counted in length value.

The DNS record will require no values, therefore the length will always be 1.

BYTE 3: High-order 8 bits of 16-bit length field. This byte will be zero for all option records.

BYTE 4: Option type value/retransmission indicator. The high-order bit (R) is set if this record is a retransmission. The low-order 7 bits represent the value for one of the fourteen currently defined option types for AUTODIN II:

value	option type
0	characteristics compatibility
1	binary mode
2	remote control of transmission and echo (RCTE)
3	go-ahead
4	extended ASCII (XASCII)
5	line width
6	page size
7	horizontal tab settings
8	vertical tab settings
9	carriage return disposition
10	linefeed disposition
11	formfeed disposition
12	horizontal tab disposition
13	vertical tab disposition

RECORD NAME: Erase Character Character
RECORD TYPE ACRONYM: EC
RECORD TYPE VALUE: 217
DESCRIPTION: The EC record is added to the to-network buffer when the local EC character is recognized in the user's data stream. The destination THP will convert the record to the destination user's EC character and send it to that user. The EC character is not an NVT character which affects the NVT printer and, therefore, is always sent in record form whether the connection is in stream or record mode.

RECORD FORMAT:

```
        15        7        0
        -------------------
        !        !        !
   1    !   EC   !   RM   !   0
        !        !        !
        -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: EC record type value (217).

RECORD NAME: Erase Line Character
RECORD TYPE ACRONYM: EL
RECORD TYPE VALUE: 218
DESCRIPTION: The EL record is added to the to-network buffer when the local EL character is recognized in the user's data stream. The destination TMP will convert the record to the destination user's EL character and send it to that user. The EL character is not an NVT character which affects the NVT printer and, therefore, is always sent in record form whether the connection is in record or stream mode.

RECORD FORMAT:

```
        15 . . . 7           0
        --------------------
        |        |         |
    1   |   EL   |   RM    |   0
        |        |         |
        --------------------
```

BYTE 0: TMP standard record mark value (170), denoting beginning of TMP record.

BYTE 1: EL record type value (218).

RECORD NAME: Formfeed Character
RECORD TYPE ACRONYM: FF
RECORD TYPE VALUE: 211
DESCRIPTION: The FF record is added to the to-network buffer when the local FF character is recognized in the user's data stream. The destination TMP will convert the record to the destination user's FF character and send it to that user. The FF character is an NVT character which affects the NVT printer and is not sent in record form when the connection is in stream mode. The NVT value (record type value) is sent in place of the local user's FF character.

RECORD FORMAT:

```
         15        7        0
         -------------------
         !        !        !
    1    !   FF   !   RM   !   0
         !        !        !
         -------------------
```

BYTE 0: TMP standard record mark value (170), denoting beginning of TMP record.

BYTE 1: FF record type value (211).

RECORD NAME: Go-Ahead
RECORD TYPE ACRONYM: GA
RECORD TYPE VALUE: 220
DESCRIPTION: The GA record is added to the to-network buffer when the local GA character is recognized in the user's data stream. The destination THP will convert the record to the destination user's GA character and send it to that user. The GA character is not an NVT character which affects the NVT printer and, therefore is always sent in record form whether the connection is in record or stream mode.

RECORD FORMAT:

```
             15        7       0
             -------------------
             |        |        |
          1  |   GA   |   RM   |  0
             |        |        |
             -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: GA record type value (220).

RECORD NAME: Horizontal Tab Character
RECORD TYPE ACRONYM: HT
RECORD TYPE VALUE: 212
DESCRIPTION: The HT record is added to the to-network buffer when the local HT character is recognized in the user's data stream. The destination THP will convert the record to the destination user's HT character and send it to that user. The HT character is an NVT character which affects the NVT printer and is not sent in record form when the connection is in stream mode. The NVT value (record type value) is sent in place of the local user's HT character.

RECORD FORMAT:

```
          15        7        0
          -------------------
          !        !        !
      1   !   HT   !   RM   !   0
          !        !        !
          -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: HT record type value (212).

RECORD NAME: Interrupt Function (IF)

RECORD TYPE ACRONYM: IFn (IF1, IF3, IF4, IF5, IF6, IF7, IF9, IF11, IF13, IF14, IF16, IF17, IF19)

RECORD TYPE VALUE: 241-253 (IF1-IF19, as listed above)

DESCRIPTION: There are thirteen IFn records defined, each causing a unique function to take place across the network. All IFn records are in the format as shown below, but each has a unique IFn value in byte 1 of the record. There are only 10 interrupt functions defined, but IF1-9 are duplicated by IF11-19, giving the user two characters at either end (as required) of the connection for each function. IF2 causes a source to destination flush with no signal to the user, therefore, does not require a THP record. IF5, 8, 15, and 18 do not require a signal to the user and are, therefore, sent to the destination THP via the IF5 record. IF10 does not require a THP record as it is handled via the TCP out-of-band interrupt function.

The IFn record is added to the to-network buffer when the local IFn character is recognized in the user's data stream. The to-network buffer will then be sent. The destination THP will convert the record to the destination user's IFn character and send it to that user, as appropriate. The destination THP will cause the requested flush function upon recognition of the appropriate IFn records. The IFn characters are not NVT characters which affect the NVT printer and, therefore, are always sent in record form whether the connection is in record or stream mode.

RECORD FORMAT:

```
        15         7         0
        -------------------
        |        |        |
    1   |  IFn   |   RM   |   0
        |        |        |
        -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: IF1, IF3, IF4, IF5, IF6, IF7, IF9, IF11, IF13, IF14, IF16, IF17, or IF19 record type value (241-253, respectively).

RECORD NAME: Linefeed Character
RECORD TYPE ACRONYM: LF
RECORD TYPE VALUE: 212
DESCRIPTION: The LF record is added to the to-network buffer when the local LF character is recognized in the user's data stream. The destination TMP will convert the record to the destination user's LF character and send it to that user. The LF character is an NVT character which affects the NVT printer and is not sent in record form when the connection is in stream mode. The NVT value (record type value) is sent in place of the local user's LF character.

RECORD FORMAT:

```
        15 . . . 7         0
        --------------------
        !        !         !
    1   !   LF   !   RM    !   0
        !        !         !
        --------------------
```

BYTE 0: TMP standard record mark value (170), denoting beginning of TMP record.

BYTE 1: LF record type value (212).

RECORD NAME: Null Character
RECORD TYPE ACRONYM: NUL
RECORD TYPE VALUE: 216
DESCRIPTION: The NUL record is added to the to-network buffer when the local NUL character is recognized in the user's data stream. The destination THP will convert the record to the destination user's NUL character and send the appropriate number of nulls to the user, based on the count in the NUL record. The NUL character is an NVT character which affects the NVT printer and is not sent in record form when the connection is in stream mode. The NVT value (record type value) is sent in place of the local user's NUL character.

RECORD FORMAT:

```
        15         7         0
        -------------------
        !        !        !
    1   !  NUL   !   RM   !   0
        !        !        !
        -------------------
        !        !        !
    3   ! -----  ! count  !   2
        !        !        !
        -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: NUL record type value (216).

BYTE 2: Number of nulls to be sent to the destination user (1-255).

RECORD NAME: Remote Control of Transmission and Echo Command

RECORD TYPE ACRONYM: RCT

RECORD TYPE VALUE: 231

DESCRIPTION: The RCT record is added to the to-network buffer when the user enters an RCTE command. Of course, the RCTE command is not valid unless the RCTE option has been successfully negotiated with the remote THP. The user designates, via RCTE command parameters, the new echo mode or packet release mechanism for the destination terminal user. Only one of these need be designated by the user, but both may be changed, if desired.

RECORD FORMAT:

```
       15        7        0
       -------------------
       |        |        |
   1   |  RCT   |   RM   |   0
       |        |        |
       -------------------
       |        |        |
   3   |     length      |   2
       |        |        |
       -------------------
       |        |        |
   5   |  char  |  info  |   4
       |        |        |
       -------------------
       |        |        |
   7   | -----  |  char  |   6
       |        |        |
       -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: RCT record type value (231).

BYTE 2: Low-order 8 bits of 16-bit length value. Length is a value representing number of bytes in the RCT record beyond the length field. For an RCT record, byte 2 will contain three (3).

BYTE 3: High-order 8 bits of 16-bit length value. Length is a value representing number of bytes in the RCT record beyond the length field. For an RCT record, byte 3 will always be zero (0).

BYTE 4: RCTE command information, designating action to be taken by destination TMP. The bits and meaning of each (when set) is:

bit	meaning
0	echo change reflected in bit 1
1	if set, turn echo on; if clear, turn echo off
2	if set, make packet release mechanism "on number of characters," (reflected in bytes 5 and 6)
3	if set, make packet release mechanism "on new line sequence"
4	if set, make packet release mechanism "on send now character"
5	if set, make packet release mechanism "as received"
6	--not used--
7	--not used--

BYTE 5: Low-order 8 bits of 16-bit value representing number of characters to accumulate before releasing telnet buffer. Bit 2 of byte 4 will be set, indicating that the packet release mechanism for the terminal user should be changed to "on specified number of characters."

BYTE 6: High-order 8 bits of 16-bit value representing number of characters to accumulate before releasing telnet buffer.

```
      RECORD NAME:  Request Record Mode
RECORD TYPE ACRONYM:  RQRM
  RECORD TYPE VALUE:  229
        DESCRIPTION:  The RQRM record is not sent by  CCU/TAC  TMPs,
                      but  will  be  recognized by them, and honored
                      (Set Record Mode record returned),  if  appro-
                      priate  for the user and the particular imple-
                      mentation.  In general, a request to enter re-
                      cord  mode will always be honored by a CCU/TAC
                      TMP.

    RECORD FORMAT:

                      15       7       0
                      ------------------
                      |       |        |
                  1   | RQRM  |   RM   |  2
                      |       |        |
                      ------------------

          BYTE 0:  TMP standard record  mark  value  (170),
                   denoting beginning of TMP record.

          BYTE 1:  RQRM record type value (229).
```

```
          RECORD NAME:  Request Stream Mode
   RECORD TYPE ACRONYM:  RQSM
     RECORD TYPE VALUE:  230
           DESCRIPTION:  The RQSM record will not be  sent  by  CCU/TAC
                         TMPs, but will be recognized by them.  In gene-
                         ral, CCU/TAC TMPs will respond by  sending  a
                         Set Record Mode record.  That is, CCU/TAC TMPs
                         will enter stream mode only when  so  directed
                         by the local user, via the mode command.

        RECORD FORMAT:

                          15        7       0
                          ---------------------
                          |        |         |
                      1   |  RQSM  |   RM    |   0
                          |        |         |
                          ---------------------


           BYTE 0:  TMP standard record  mark  value  (177),
                    denoting beginning of TMP record.

           BYTE 1:  RQSM record type value (230).
```

RECORD NAME: Set Record Mode
RECORD TYPE ACRONYM: SRM
RECORD TYPE VALUE: 227
DESCRIPTION: The SRM record will be added to the to-network buffer when the user has entered a mode command, changing the current method of packaging data to record mode. This record will tell the destination TNP that network data will now be in record mode. The SRM record will also be sent in response to either request mode record (RQRM or RQSM), announcing to the destination TNP that the connection is in record mode.

RECORD FORMAT:

```
          15        7         0
          -------------------
          |        |        |
     1    |  SRM   |   RM   |   0
          |        |        |
          -------------------
```

BYTE 0: TNP standard record mark value (170), denoting beginning of TNP record.

BYTE 1: SRM record type value (227).

RECORD NAME: Set Stream Mode
RECORD TYPE ACRONYM: SSM
RECORD TYPE VALUE: 228
DESCRIPTION: The SSM record will be added to the to-network buffer when the user has entered a mode command, changing the current method of packaging data to stream mode. This will notify the destination TMP that the method being used to package network data is now stream mode. The SSM record will also be sent in response to either request mode record (RQRM or RQSM) if the connection is in stream mode, announcing the current packaging mode.

RECORD FORMAT:

```
              15         7         0
              --------------------
              !         !         !
          1   !   SSM   !   RM    !   0
              !         !         !
              --------------------
```

BYTE 0: TMP standard record mark value (172), denoting beginning of TMP record.

BYTE 1: SSM record type value (228).

RECORD NAME: Status Request
RECORD TYPE ACRONYM: STS
RECORD TYPE VALUE: 225
DESCRIPTION: The STS record is added to the to-network buffer when the user enters a status command and the local (source) CCU/TAC seems to be clear. That is, the user has an active connection and neither THP nor TCP is in a hold off condition. The STS record is sent to the destination THP, both to ensure that the network is functioning and to ensure that the remote (destination) access area (CCU/TAC) is functioning. The STS record will be answered with a Status Reply record, if the destination THP is intact.

RECORD FORMAT:

```
        15          7         0
        ---------------------
        !          !         !
    1   !   STS    !   RM    !   0
        !          !         !
        ---------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: STS record type value (225).

RECORD NAME: Status Reply
RECORD TYPE ACRONYM: SRP
RECORD TYPE VALUE: 226
DESCRIPTION: The SRP record will be added to the to-network buffer when TMP recognizes the Status Request record in the network data stream. The SRP record will signify that TMP and, consequently, the access area (CCU/TAC), are functioning.

RECORD FORMAT:

```
             15         7         0
             ----------------------
             |          |         |
        1    |   SRP    |   RM    |   0
             |          |         |
             ----------------------
```

BYTE 0: TMP standard record mark value (170), denoting beginning of TMP record.

BYTE 1: SRP record type value (226).

RECORD NAME: Vertical Tab Character
RECORD TYPE ACRONYM: VT
RECORD TYPE VALUE: 213
DESCRIPTION: The VT record is added to the to-network buffer when the local VT character is recognized in the user's data stream. The destination TNP will convert the record to the destination user's VT character and send it to that user. The VT character is an NVT character which affects the NVT printer and is not sent in record form when the connection is in stream mode. The NVT value (record type value) is sent in place of the local user's VT character.

RECORD FORMAT:

```
      15        7       0
      --------------------
      !         !        !
   1  !   VT    !   RM   !  0
      !         !        !
      --------------------
```

BYTE 0: TNP standard record mark value (170), denoting beginning of TNP record.

BYTE 1: VT record type value (213).

RECORD NAME: Will Receiver Option Negotiation Record
RECORD TYPE ACRONYM: WLR
RECORD TYPE VALUE: 239
DESCRIPTION: The WLR option negotiation record is added to the to-network buffer when the local user enters an option negotiation command, or during open processing when the characteristics option is negotiated. The detailed description of each option type record is shown in Appendix B. This shows the general format for an WLR option negotiation record, without regard to option type.

The WLR option negotiation record indicates that the specified option should be started on the requester's receive side.

The WLR record is used for those option types which require that both THPs perform some processing, such as, characteristics, binary mode, XASCII, RCTE, and go-ahead.

RECORD FORMAT:

```
        15       7        0
        --------------------
        !        !        !
   1    !  WLR   !   RM   !   0
        !        !        !
        --------------------
        !        !        !
   3    !      length     !   2
        !        !        !
        --------------------
        !        ! !      !
   5    ! param  !R! type !   4
        !        ! !      !
        --------------------
        !        !        !
   7    ! param  ! param  !   6
        !        !        !
        --------------------
        !        !        !
   9    ! param  ! param  !   8
        !        !        !
        --------------------
        !        !        !
  11    ! ------ ! param  !  10
        !        !        !
        --------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: WLR record type value (239).

BYTE 2: Low-order 8 bits of 16-bit length field. Length value represents number of bytes in the record beyond the length field, i.e., first four bytes of the record are not counted in length value.

An option negotiation record will not be longer than 38 bytes total i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 3: High-order 8 bits of 16-bit length field. Because option records will be no longer than 38 bytes, this byte will always be zero for option records.

BYTE 4: Option type value/retransmission indicator. The high-order bit (R) is set if this record is a retransmission. The low-order 7 bits represent the value for one of the fourteen currently defined option types for AUTODIN II:

value	option type
0	characteristics compatibility
1	binary mode
2	remote control of transmission and echo (RCTE)
3	go-ahead
4	extended ASCII (XASCII)
5	line width
6	page size
7	horizontal tab settings
8	vertical tab settings
9	carriage return disposition
10	linefeed disposition
11	formfeed disposition
12	horizontal tab disposition
13	vertical tab disposition

BYTES 5 thru 10: Additional parameter bytes for some options, such as tab settings or line width/page size value.

RECORD NAME: Won't Receiver Option Negotiation Record
RECORD TYPE ACRONYM: WNR
RECORD TYPE VALUE: 240
DESCRIPTION: The WNR option negotiation record is added to the to-network buffer when the local user enters an option negotiation command, or during open processing when the characteristics option is negotiated. The detailed description of each option type record is shown in Appendix B. This shows the general format for an WNR option negotiation record, without regard to option type.

The WNR option negotiation record indicates that the specified option should be stopped on the requester's receive side.

RECORD FORMAT:

```
      15         7         0
      ---------------------
      |         |         |
  1   |   WNR   |   RM    |   0
      |         |         |
      ---------------------
      |         |         |
  3   |      length       |   2
      |         |         |
      ---------------------
      |         | |       |
  5   |  -----  |R| type  |   4
      |         | |       |
      ---------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: WNR record type value (240).

BYTE 2: Low-order 8 bits of 16-bit length field. Length value represents number of bytes in the record beyond the length field, i.e., first four bytes of the record are not counted in length value.

The WNR record will require no values, therefore the length will always be 1.

BYTE 3: High-order 8 bits of 16-bit length field. Because option records will be no longer than 30 bytes, this byte will always be zero for option records.

BYTE 4: Option type value/retransmission indicator. The high-order bit (R) is set if this record is a retransmission. The low-order 7 bits represent the value for one of the fourteen currently defined option types for AUTODIN II:

value	option type
0	characteristics compatibility
1	binary mode
2	remote control of transmission and echo (RCTE)
3	go-ahead
4	extended ASCII (XASCII)
5	line width
6	page size
7	horizontal tab settings
8	vertical tab settings
9	carriage return disposition
10	linefeed disposition
11	formfeed disposition
12	horizontal tab disposition
13	vertical tab disposition

RECORD NAME: Will Sender Option Negotiation Record
RECORD TYPE ACRONYM: WLS
RECORD TYPE VALUE: 237
DESCRIPTION: The WLS option negotiation record is added to the to-network buffer when the local user enters an option negotiation command, or during open processing when the characteristics option is negotiated. The detailed description of each option type record is shown in Appendix B. This shows the general format for an WLS option negotiation record, without regard to option type.

The WLS option negotiation record indicates that the specified option should be started on the requester's receive side.

RECORD FORMAT:

```
          15       7        0
          -------------------
          !        !        !
     1    !  WLS   !   RM   !   0
          !        !        !
          -------------------
          !        !        !
     3    !    length       !   2
          !        !        !
          -------------------
          !        ! !      !
     5    !  param !R! type !   4
          !        ! !      !
          -------------------
          !        !        !
     7    !  param ! param  !   6
          !        !        !
          -------------------
          !        !        !
     9    !  param ! param  !   8
          !        !        !
          -------------------
          !        !        !
    11    !  ----- ! param  !   10
          !        !        !
          -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: WLS record type value (237).

BYTE 2: Low-order 8 bits of 16-bit length field. Length value represents number of bytes in the record beyond the length field, i.e., first four bytes of the record are not counted in length value.

An option negotiation record will not be longer than 38 bytes total i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 3: High-order 8 bits of 16-bit length field. Because option records will be no longer than 38 bytes, this byte will always be zero for option records.

BYTE 4: Option type value/retransmission indicator. The high-order bit (R) is set if this record is a retransmission. The low-order 7 bits represent the value for one of the fourteen currently defined option types for AUTODIN II:

value	option type
0	characteristics compatibility
1	binary mode
2	remote control of transmission and echo (RCTE)
3	go-ahead
4	extended ASCII (XASCII)
5	line width
6	page size
7	horizontal tab settings
8	vertical tab settings
9	carriage return disposition
10	linefeed disposition
11	formfeed disposition
12	horizontal tab disposition
13	vertical tab disposition

BYTES 5 thru 10: Additional parameter bytes for some options, such as tab settings or line width/page size value.

RECORD NAME: won't Sender Option Negotiation Record
RECORD TYPE ACRONYM: WNS
RECORD TYPE VALUE: 238
DESCRIPTION: The WNS option negotiation record is added to the to-network buffer when the local user enters an option negotiation command, or during open processing when the characteristics option is negotiated. The detailed description of each option type record is shown in Appendix B. This shows the general format for an WNS option negotiation record, without regard to option type.

The WNS option negotiation record indicates that the specified option should be stopped on the requester's receive side.

RECORD FORMAT:

```
             15        7        0
              --------------------
              |        |         |
          1   |  WNS   |   RM    |   0
              |        |         |
              --------------------
              |        |         |
          3   |     length       |   2
              |        |         |
              --------------------
              |        | |       |
          5   | -----  |R| type  |   4
              |        | |       |
              --------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: WNS record type value (238).

BYTE 2: Low-order 8 bits of 16-bit length field. Length value represents number of bytes in the record beyond the length field, i.e., first four bytes of the record are not counted in length value.

The WNS record will require no values, therefore the length will always be 1.

BYTE 3: High-order 8 bits of 16-bit length field. Because option records will be no longer than 38 bytes, this byte will always be zero for option records.

BYTE 41: Option type value/retransmission indicator. The high-order bit (R) is set if this record is a retransmission. The low-order 7 bits represent the value for one of the fourteen currently defined option types for AUTODIN II:

value	option type
0	characteristics compatibility
1	binary mode
2	remote control of transmission and echo (RCTE)
3	go-ahead
4	extended ASCII (XASCII)
5	line width
6	page size
7	horizontal tab settings
8	vertical tab settings
9	carriage return disposition
10	linefeed disposition
11	formfeed disposition
12	horizontal tab disposition
13	vertical tab disposition

RECORD NAME: XASCII Record

RECORD TYPE ACRONYM: XA

RECORD TYPE VALUE: 222

DESCRIPTION: The XA record is added to the to-network buffer when the user-to-network data is being scanned in XASCII mode. The XA record does not contain the XASCII shift-out (SO) or shift-in (SI) character. These characters must be inserted by the destination TMP upon entering and leaving XASCII mode, respectively. The first XA record will indicate that XASCII mode is beginning. The last consecutive XA record will indicate that XASCII mode is ending and that the destination user's local SI character value must be sent in the to-user buffer.

RECORD FORMAT:

```
          15        7        0
          --------------------
          !        !         !
     1    !   XA   !   RM    !   0
          !        !         !
          --------------------
          !        !         !
     3    !     length       !   2
          !        !         !
          --------------------
          !        !         !
     5    ! octet  ! octet   !   4
     .    !        !         !   .
     .    --------------------   .
     .    !        !         !   .
     n    !        !         !  n-1
          !        !         !
          --------------------
```

BYTE 0: TMP standard record mark value (172), denoting beginning of TMP record.

BYTE 1: XA record type value (222).

BYTE 2: Low-order 8 bits of 16-bit length value, representing number of bytes in the record beyond the length field, i.e., the first four bytes of the record are not counted in the length value.

BYTE 3: High-order 3 bits of 16-bit length value, representing number of bytes in the record beyond the length field, i.e., the first four bytes of the record are not counted in the length value.

BYTES 4 thru "n": XASCII data, ready to be sent to the destination user. XASCII data consists of octets which are put in the XA record without stripping the parity bit or any other processing, other than checking for the terminating SI character.

APPENDIX B

OPTION NEGOTIATION RECORDS

This appendix describes the THP-THP option negotiation record format to be used by a THP desiring to interface with and negotiate defined options with CCU/TAC THPs on the AUTODIN II network. There is no variation to these formats as described herein.

There are currently 14 options defined for AUTODIN II. The characteristics option is unique in that there is only a request to "start" the option. Because the option processing is one-time only, there is no requirement for a request to stop the characteristics option. All of the other options can be started and stopped for either side of the connection and with either THP performing the option processing. CCU/TAC THPs, however, will not agree to any option request which will result in a CCU/TAC THP performing non-NVT processing on its user's send side, i.e., CCU/TAC THPs will not send DOS records.

The general format for the option records is described in Appendix A, so it is not repeated here. Included in the description in this appendix for each record is the option name, accepted acronym, option type value (decimal), a brief description of the record and its purpose, and the format for the record.

The format is shown in standard DEC PDP-11 memory image format, with the least significant byte on the right (even number) and the most significant byte on the left (odd number) of a 16-bit word. The bits are labeled 0 to 15 (least significant to most significant) right to left in the word. The direction of transmission is from the least significant byte/least significant bit (0) to the most significant byte/most significant bit (15) for each word.

The memory image for each record shown in this appendix is described by an accompanying narrative. The size (byte, word, etc.) of each unit within the structure may cause an "extra" byte at the end of the record. This extra byte is not used and will not be defined in the narrative. It will be represented by a series of dashes (- - -) in the memory image. The extra byte will not be counted in the length of the structure and in fact is not sent as part of the record.

The records are listed in alphabetical order by the option type name, as referenced in the text of this document.

OPTION NAME: Binary Mode Option
OPTION TYPE ACRONYM: BIN
OPTION TYPE VALUE: 1
DESCRIPTION: The BIN option record is added to the to-network buffer when a user enters an option command requesting to start or stop the BIN option or in response to an option record from the remote TMP requesting that the BIN option be started or stopped.

The BIN option requires the use of the receive-oriented option record types, e.g., DOR, DNR, WLR, or WNR.

Both the request to start (DOR or WLR) and the request to stop (DNR or WNR) the BIN option are in the format as described below.

RECORD FORMAT:

```
     15        7         0
     --------------------
     !         !         !
  1  ! Rcdtyp  !   RM    !  0
     !         !         !
     --------------------
     !         !         !
  3  !    0    !    1    !  2
     !         !         !
     --------------------
     !         !         !
  5  !  -----  !   BIN   !  4
     !         !         !
     --------------------
```

BYTE 0: TMP standard record mark value (172), denoting beginning of TMP record.

BYTE 1: Option record type value, one of the four option record type values described in Appendix A for the option records DOR, DNR, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the BIN option type, value in this byte is 1.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 35 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: SIN option type value (1).

OPTION NAME: Carriage Return Disposition Option
OPTION TYPE ACRONYM: CRD
OPTION TYPE VALUE: 9
DESCRIPTION: The CRD option record is added to the to-network buffer when a user enters an option command requesting to start or stop the CRD option or in response to an option record from the remote TMP requesting that the CRD option be started or stopped.

The CRD option record need not contain the "value" field shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

```
     15         7         0
     -------------------
     !        !        !
  1  ! rcdtyp !   RM   !   0
     !        !        !
     -------------------
     !        !        !
  3  !   0    !   2    !   2
     !        !        !
     -------------------
     !        !        !
  5  ! value  !  CRD   !   4
     !        !        !
     -------------------
```

BYTE 0: TMP standard record mark value (170), denoting beginning of TMP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, WLS, WNS, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the CRD option type, value in this byte is 2 if the record is sent as a request to start the CRD option. The value is 1, if the record is sent as a request to stop the CRD option and does not contain the "value" byte.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: CRD option type value (9).

BYTE 5: Carriage return disposition indicator, designating requested disposition for a carriage return. This byte may be omitted for a request to stop a previously negotiated option, but the length field must be adjusted, as described above. The carriage return disposition indicator will be one of the following values:

value	disposition
0-253	insert specified number of time delays (ASCII nulls) after carriage return
254	discard carriage return
255	deliver carriage return as received

OPTION NAME: Characteristics Option
OPTION TYPE ACRONYM: CHR
OPTION TYPE VALUE: 0
DESCRIPTION: The CHR option record is added to the to-network buffer during open processing and is always sent as a PLR option request. The source THP is requesting that binary mode be started for the side of the connection on which data comes toward his user (receive side), if the data being sent by the destination THP's user is in the format required by the source user's characteristics. These are the characteristics sent in the CHR option record.

In responding to the CHR option, the destination THP will return the CHR record as received except for the option record type value. The option record type value will be either OCR or ONR, depending on the outcome of the characteristics compatibility checking.

RECORD FORMAT:

```
       15        7       0
       ---------------------
       |         |         |
    1  | rcdtyp  |   RM    |  0
       |         |         |
       ---------------------
       |         |         |
    3  |    0    |   34    |  2
       |         |         |
       ---------------------
       |         |         |
    5  | contyp  |   CHR   |  4
       |         |         |
       ---------------------
       |         |         |
    7  |  goval  | go-reg  |  6
       |         |         |
       ---------------------
       |         |         |
    9  |  ps     |  lw     |  8
       |         |         |
       ---------------------
       |         |         |
   11  |  erdis  |  erval  |  10
       |         |         |
       ---------------------
```

```
     ---------------------
     |         |         |
  13 |  lfdis  |  lfval  |  12
     |         |         |
     ---------------------
     |         |         |
  15 |  ffdis  |  ffval  |  14
     |         |         |
     ---------------------
     |         |         |
  17 |  htdis  |  htval  |  16
     |         |         |
     ---------------------
     |         |         |
  19 |  vtdis  |  vtval  |  18
     |         |         |
     ---------------------
     |         |         |
  21 |  ecval  | nulval  |  20
     |         |         |
     ---------------------
     |         |         |
  23 |  bsval  |  elval  |  22
     |         |         |
     ---------------------
     |         |         |
  25 | newlin  | belval  |  24
     |         |         |
     ---------------------
     |         |         |
  27 |    horizontal     |  26
     |                   |
     --                 --
     |                   |
  29 |        tab        |  28
     |                   |
     --                 --
     |                   |
  31 |     settings      |  30
     |         |         |
     ---------------------
     |         |         |
  33 |     vertical      |  32
     |                   |
     --                 --
     |                   |
  35 |        tab        |  34
     |                   |
     --                 --
     |                   |
  37 |     settings      |  36
     |         |         |
     ---------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: Option record type value for ONR, DOR, or DNR, as appropriate. ONR is always sent as the request. DOR/DNR is response, as appropriate.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the CHR option type, value in this byte is 34.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: CHR option type value (2).

BYTE 5: Cross-connection terminal type, used in determining if connection is valid according to the approved cross-connection matrix.

BYTE 6: Go-ahead requirements, used to determine if go-ahead processing is compatible for the two users. If local user allows negotiation of the go-ahead option on his receive side, the byte value is one. If not, the byte value is zero.

BYTE 7: Go-ahead character value.

BYTE 8: Line width, maximum number of characters to be put on one line of user's "printer."

BYTE 9: Page size, maximum number of lines to be put on one page of user's "printer."

BYTE 10: Carriage return character value.

BYTE 10: Carriage return disposition, user's default disposition for carriage return.

BYTE 12: Linefeed character value.

BYTE 13: Linefeed disposition, user's default disposition for linefeed.

BYTE 14: Formfeed character value.

BYTE 15: Formfeed disposition, user's default disposition for formfeed.

BYTE 16: Horizontal tab character value.

BYTE 17: Horizontal tab disposition, user's default disposition for horizontal tab.

BYTE 18: Vertical tab character value.

BYTE 19: Vertical tab disposition, user's default disposition for vertical tab.

BYTE 20: Null character value.

BYTE 21: Erase character character value.

BYTE 22: Erase line character value.

BYTE 23: Backspace character value.

BYTE 24: Bell character value.

BYTE 25: New line type, indicating whether user requires a carriage return/linefeed sequence (byte is 0), or carriage return only (byte is 1) for the end-of-line sequence.

BYTES 26 thru 31: Horizontal tab settings, one of two formats: (1) each byte contains a value representing a tab setting, these must be in ascending order; or (2) first byte contains 255 (decimal) and second byte contains a value representing the first tab setting and the increment to each subsequent tab settings.

BYTES 32 thru 37: Vertical tab settings, one of two formats: (1) each byte contains a value representing a tab setting, these must be in ascending order; or (2) first byte contains 255 (decimal) and second byte contains a value representing the first tab setting and the increment to each subsequent tab settings.

OPTION NAME: Formfeed Disposition Option
OPTION TYPE ACRONYM: FFD
OPTION TYPE VALUE: 11
DESCRIPTION: The FFD option record is added to the to-network buffer when a user enters an option command requesting to start or stop the FFD option or in response to an option record from the remote TMP requesting that the FFD option be started or stopped.

The FFD option record need not contain the "value" field shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

```
      15       7       0
      -------------------
      !        !        !
  1   ! recdtyp!   RM   !   0
      !        !        !
      -------------------
      !        !        !
  3   !   0    !   2    !   2
      !        !        !
      -------------------
      !        !        !
  5   ! value  !  FFD   !   4
      !        !        !
      -------------------
```

BYTE 0: TMP standard record mark value (172), denoting beginning of TMP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, RLS, RNS, RLR, and RNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the FFD option type, value in this byte is 2 if the record is sent as a request to start the FFD option. The value is 1, if the record is sent as a request to stop the FFD option and does not contain the "value" byte.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: FFD option type value (11).

BYTE 5: Formfeed disposition indicator, designating requested disposition for a formfeed. This byte may be omitted for a request to stop a previously negotiated option, but the length field must be adjusted, as described above. The formfeed disposition indicator will be one of the following values:

value	disposition
0-25?	insert specified number of time delays (ASCII nulls) after formfeed
252	replace formfeed with end-of-line function
253	simulate formfeed function
254	discard formfeed
255	deliver formfeed as received

OPTION NAME: Go-ahead Option
OPTION TYPE ACRONYM: GOA
OPTION TYPE VALUE: 3
DESCRIPTION: The GOA option record is added to the to-network buffer when a user enters an option command requesting to start or stop the GOA option or in response to an option record from the remote THP requesting that the GOA option be started or stopped.

The GOA option requires the use of the receive-oriented option record types, e.g., DOR, DNR, WLR, or WNR.

Both the request to start (DOR or WLR) and the request to stop (DNR or WNR) the GOA option are in the format as described below.

RECORD FORMAT:

```
         15         7        0
         --------------------
         |        |         |
      1  | Rcdtyp |   RM    |  0
         |        |         |
         --------------------
         |        |         |
      3  |   0    |    1    |  2
         |        |         |
         --------------------
         |        |         |
      5  | -----  |   GOA   |  4
         |        |         |
         --------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: Option record type value, one of the four option record type values described in Appendix A for the option records DOR, DNR, WLR, and WNR.

BYTE 2: order 8 bits of 16-bit length field. the case of the GOA option type, in this byte is 1.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: SOA option type value (3).

OPTION NAME: Horizontal Tab Stops Option
OPTION TYPE ACRONYM: HT
OPTION TYPE VALUE: 7
DESCRIPTION: The HT option record is added to the to-network buffer when a user enters an option command requesting to start or stop the HT option or in response to an option record from the remote TNP requesting that the HT option be started or stopped.

The HT option record need not contain the "value" fields shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

	15	7 0	
1	Rectyp	RM	0
3	0	7	2
5	value	HT	4
7	value	value	6
9	value	value	8
11	-----	value	10

BYTE 0: TNP standard record mark value (170), denoting beginning of TNP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, RLS, RNS, RLR, and RNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the HT option type, value in this byte has range of 2-7 if the record is sent as a request to start the HT option. The actual length depends on the number of parameters included in the record. The value is 1, if the record is sent as a request to stop the HT option and does not contain the "value" bytes.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: HT option type value (7).

BYTES 5 thru 17: Requested horizontal tab settings, one of two formats: (1) each byte contains a value representing a tab setting, these must be in ascending order, or (2) first byte containing a 255 (decimal) and second byte contains a value which represents not only the first tab setting but also the incremental value to subsequent tab settings (i.e., one value is used for both parameters).

OPTION NAME: Horizontal Tab Disposition Option
OPTION TYPE ACRONYM: HTD
OPTION TYPE VALUE: 12
DESCRIPTION: The HTD option record is added to the to-network buffer when a user enters an option command requesting to start or stop the HTD option or in response to an option record from the remote THP requesting that the HTD option be started or stopped.

The HTD option record need not contain the "value" field shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

```
        15       7       0
        -----------------
        |       |       |
     1  | rectype|  RM   |  0
        |       |       |
        -----------------
        |       |       |
     3  |   0   |   2   |  2
        |       |       |
        -----------------
        |       |       |
     5  | value |  HTD  |  4
        |       |       |
        -----------------
```

BYTE 0: THP standard record mark value (172), denoting beginning of THP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, WLS, WNS, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the HTD option type, value in this byte is 2 if the record is sent as a request to start the HTD option. The value is 1, if the record is sent as a request to stop the HTD option and does not contain the "value" field.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than six (34).

BYTE 4: HTD option type value (12).

BYTE 5: horizontal tab disposition indicator, designating requested disposition for a horizontal tab. This byte may be omitted for a request to stop a previously negotiated option, but the length field must be adjusted, as described above. The horizontal tab disposition indicator will be one of the following values:

value	disposition
0-252	insert specified number of time delays (ASCII nulls) after horizontal tab
253	simulate horizontal tab function
254	discard horizontal tab
255	deliver horizontal tab as received

OPTION NAME: Linefeed Disposition Option
OPTION TYPE ACRONYM: LFD
OPTION TYPE VALUE: 10
DESCRIPTION: The LFD option record is added to the to-network buffer when a user enters an option command requesting to start or stop the LFD option or in response to an option record from the remote TNP requesting that the LFD option be started or stopped.

The LFD option record need not contain the "value" field shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

```
         15        7         0
         ---------------------
         |         |         |
    1    | rectyp  |   RM    |  0
         |         |         |
         ---------------------
         |         |         |
    3    |    0    |    2    |  2
         |         |         |
         ---------------------
         |         |         |
    5    |  value  |   LFD   |  4
         |         |         |
         ---------------------
```

BYTE 0: TNP standard record mark value (172), denoting beginning of TNP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, WLS, WNS, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the LFD option type, value in this byte is 2 if the record is sent as a request to start the LFD option. The value is 1, if the record is sent as a request to stop the LFD option and does not contain the "value" byte.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: LFD option type value (12).

BYTE 5: Linefeed disposition indicator, designating requested disposition for a linefeed. This byte may be omitted for a request to stop a previously negotiated option, but the length field must be adjusted, as described above. The linefeed disposition indicator will be one of the following values:

value	disposition
0-252	insert specified number of time delays (ASCII nulls) after linefeed
253	simulate linefeed function
254	discard linefeed
255	deliver linefeed as received

OPTION NAME: Line Width Option
OPTION TYPE ACRONYM: LW
OPTION TYPE VALUE: 5
DESCRIPTION: The LW option record is added to the to-network buffer when a user enters an option command requesting to start or stop the LW option or in response to an option record from the remote TNP requesting that the LW option be started or stopped.

The LW option record need not contain the "value" field shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

```
     15        7       0
     -------------------
     !        !        !
  1  ! rcdtyp !   RM   !  0
     !        !        !
     -------------------
     !        !        !
  3  !   0    !   2    !  2
     !        !        !
     -------------------
     !        !        !
  5  ! value  !   LW   !  4
     !        !        !
     -------------------
```

BYTE 0: TNP standard record mark value (170), denoting beginning of TNP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, WLS, WNS, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the LW option type, value in this byte is 2 if the record is sent as a request to start the LW option. The value is 1, if the record is sent as a request to stop the LW option and does not contain the "value" byte.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: LW option type value (5).

BYTE 5: Line width indicator, designating the requested handling of line width. This byte may be omitted for a request to stop a previously negotiated option, but the length field must be adjusted, as described above. The line width indicator will be one of the following values:

value	meaning
0	ignore line width
1-250	limit number of characters per line to specified number of characters

OPTION NAME: Page Size Option
OPTION TYPE ACRONYM: PS
OPTION TYPE VALUE: 6
DESCRIPTION: The PS option record is added to the to-network buffer when a user enters an option command requesting to start or stop the PS option or in response to an option record from the remote THP requesting that the PS option be started or stopped.

The PS option record need not contain the "value" field shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

```
      15         7        0
      ---------------------
      |          |        |
   1  | rectype  |   RM   |  0
      |          |        |
      ---------------------
      |          |        |
   3  |    0     |   2    |  2
      |          |        |
      ---------------------
      |          |        |
   5  |  value   |   PS   |  4
      |          |        |
      ---------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, WLS, WNS, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the PS option type, value in this byte is 2 if the record is sent as a request to start the PS option. The value is 1, if the record is sent as a request to stop the PS option and does not contain the "value" byte.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: PS option type value (6).

BYTE 5: Page size indicator, designating requested handling of page size. This byte may be omitted for a request to stop a previously negotiated option, but the length field must be adjusted, as described above. The page size indicator will be one of the following values:

value	meaning
0	ignore page size
1-258	limit number of lines per page to specified number of lines

OPTION NAME: Remote Control of Transmission and Echo Option
OPTION TYPE ACRONYM: RCT
OPTION TYPE VALUE: 2
DESCRIPTION: The RCT option record is added to the to-network buffer when a user enters an option command requesting to start or stop the RCT option or in response to an option record from the remote TNP requesting that the RCT option be started or stopped.

The RCT option requires the use of the receiver-oriented option record types, e.g., DOR, DNR, WLR, or WNR.

Both the request to start (DOR or WLR) and the request to stop (DNR or WNR) the RCT option are in the format as described below.

RECORD FORMAT:

```
      15         7        0
      --------------------
      !         !        !
   1  ! Rectype !   RM   !  0
      !         !        !
      --------------------
      !         !        !
   3  !    0    !   1    !  2
      !         !        !
      --------------------
      !         !        !
   5  !  -----  !  RCT   !  4
      !         !        !
      --------------------
```

BYTE 0: TNP standard record mark value (170), denoting beginning of TNP record.

BYTE 1: Option record type value, one of the four option record type values described in Appendix A for the option records DOR, DNR, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the RCT option type, value in this byte is 1.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: RCT option type value (2).

OPTION NAME: Vertical Tab Stops Option
OPTION TYPE ACRONYM: VT
OPTION TYPE VALUE: 8
DESCRIPTION: The VT option record is added to the to-network buffer when a user enters an option command requesting to start or stop the VT option or in response to an option record from the remote TNP requesting that the VT option be started or stopped.

The VT option record need not contain the "value" fields shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

```
       15          7         0
       ---------------------
       |         |         |
  1    | rectyp  |   RM    |   0
       |         |         |
       ---------------------
       |         |         |
  3    |    0    |    7    |   2
       |         |         |
       ---------------------
       |         |         |
  5    |  value  |   VT    |   4
       |         |         |
       ---------------------
       |         |         |
  7    |  value  |  value  |   6
       |         |         |
       ---------------------
       |         |         |
  9    |  value  |  value  |   8
       |         |         |
       ---------------------
       |         |         |
 11    |  -----  |  value  |   10
       |         |         |
       ---------------------
```

BYTE 0: TNP standard record mark value (170), denoting beginning of TNP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, *LS, *NS, *LR, and *NR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the VT option type, value in this byte has a range of 2-7 if the record is sent as a request to start the VT option. The actual length depends on the number of parameters included in the record. The value is 1, if the record is sent as a request to stop the VT option and does not contain the "value" bytes.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: VT option type value (8).

BYTES 5 thru 17: Requested vertical tab settings, one of two formats: (1) each byte contains a value representing a tab setting, these must be in ascending order, or (2) first byte contains a 255 (decimal) and second byte contains a value which represents not only the first tab setting but also the incremental value to subsequent tab settings (i.e., one value is used for both parameters).

OPTION NAME: Vertical Tab Disposition Option
OPTION TYPE ACRONYM: VTD
OPTION TYPE VALUE: 13
DESCRIPTION: The VTD option record is added to the to-network buffer when a user enters an option command requesting to start or stop the VTD option or in response to an option record from the remote THP requesting that the VTD option be started or stopped.

The VTD option record need not contain the "value" field shown in the format below, if the record is sent as a request to stop a previously negotiated option. In that case, the length value in byte 2 would be 1.

RECORD FORMAT:

```
        15       7        0
        -------------------
        !        !        !
    1   ! rectyp !   RM   !   0
        !        !        !
        -------------------
        !        !        !
    3   !   0    !   2    !   2
        !        !        !
        -------------------
        !        !        !
    5   ! value  !  VTD   !   4
        !        !        !
        -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: Option record type value, one of the eight option record type values described in Appendix A for the option records DOS, DNS, DOR, DNR, WLS, WNS, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the VTD option type, value in this byte is 2 if the record is sent as a request to start the VTD option. The value is 1, if the record is sent as a request to stop the VTD option and does not contain the "value" byte.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: VTD option type value (13).

BYTE 5: Vertical tab disposition indicator, designating requested disposition for a vertical tab. This byte may be omitted for a request to stop a previously negotiated option, but the length field must be adjusted, as described above. The vertical tab disposition indicator will be one of the following values:

value	disposition
0-250	insert specified number of time delays (ASCII nulls) after vertical tab
252	replace vertical tab with end-of-line function
253	simulate vertical tab function
254	discard vertical tab
255	deliver vertical tab as received

OPTION NAME: XASCII Option
OPTION TYPE ACRONYM: XAS
OPTION TYPE VALUE: 4
DESCRIPTION: The XAS option record is added to the to-network buffer when a user enters an option command requesting to start or stop the XAS option or in response to an option record from the remote THP requesting that the XAS option be started or stopped.

The XAS option requires the use of the receive-oriented option record types, e.g., DOR, DNR, WLR, or WNR.

Both the request to start (DOR or WLR) and the request to stop (DNR or WNR) the XAS option are in the format as described below.

RECORD FORMAT:

```
      15        7        0
      -------------------
      |        |        |
   1  | Rcdtyp |   RM   |  0
      |        |        |
      -------------------
      |        |        |
   3  |   0    |   1    |  2
      |        |        |
      -------------------
      |        |        |
   5  | -----  |  XAS   |  4
      |        |        |
      -------------------
```

BYTE 0: THP standard record mark value (170), denoting beginning of THP record.

BYTE 1: Option record type value, one of the four option record type values described in Appendix A for the option records DOR, DNR, WLR, and WNR.

BYTE 2: Low-order 8 bits of 16-bit length field. In the case of the XAS option type, value in this byte is 1.

BYTE 3: High-order 8 bits of 16-bit length field. In all option records this byte is 0, because an option negotiation record will not be longer than 38 bytes total, i.e., the length field will contain a value no larger than thirty-four (34).

BYTE 4: XAS option type value (4).

APPENDIX C

THP STATE DIAGRAM

This appendix contains the THP state diagram for an MCCU. The state diagram shows the various states of a user's connection and the stimulus for moving from one state to another. State diagrams for the SCCU and TAC are conceptually similar to this one, as would be a state diagram for any THP.

There are eight states which have significance in the MCCU. THP "states" shown in this appendix and discussed in Paragraph 2.1.3 of the main document are based on the MCCU implementation. It is conceivable that other implementations of THP may not require so many, or may require more states. The basic "virtual connection states" are as summarized below and shown in the state diagram, but should not be construed as the only or as required states to be realized or maintained by a specific THP implementation.

1. disconnecting - a Line Disconnect event has been received from HSI when a connection was established, causing THP to send a request for an immediate close to TCP; when in this state THP is waiting for the close processing to complete

2. disconnected - hardware link (channel) associated with the user is not active; THP is waiting for notification that the link has been initialized (Line Connect event)

3. aborting - the connection has been aborted for some reason (THP protocol error, user abort command, TCP direction); THP is waiting for abort processing to complete before going to closed state

4. closing - a close command was received from the user; THP is waiting for close processing to complete before going to closed state

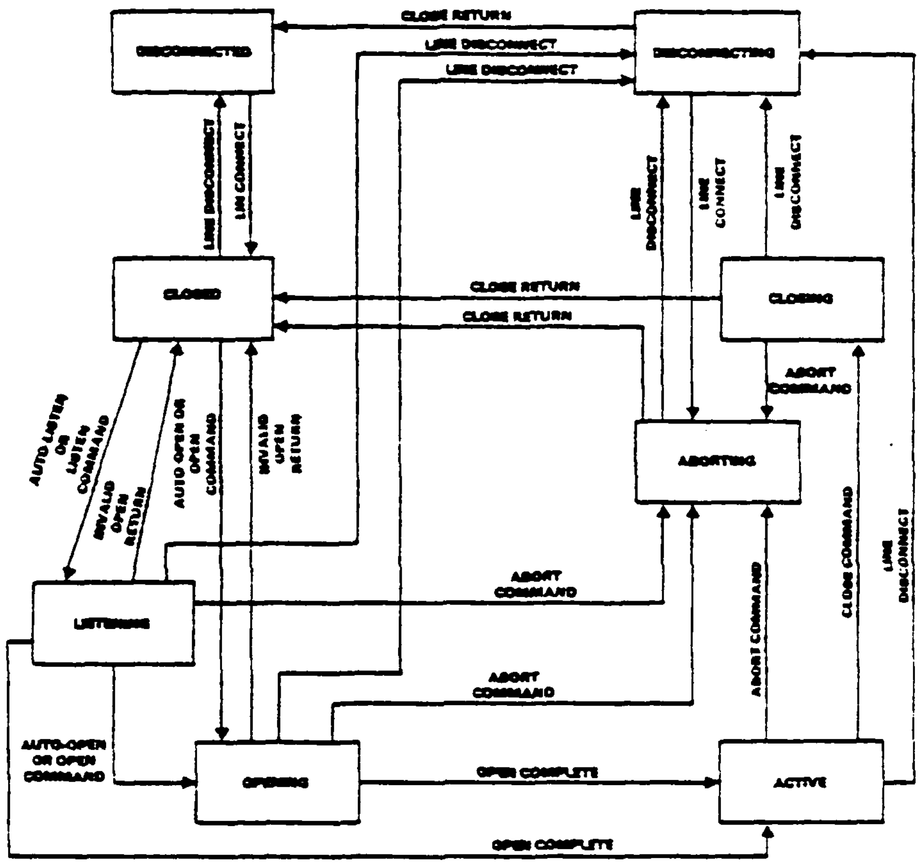
5. active - open processing is complete and connection is completely established

6. opening - a request for a connection has been received from the user or an auto-open function has been initiated; THP is waiting for notification that the connection has been established (Open Complete event); data received from the user will be saved until the connection is established and then processed as normal

7. listening - a notice has been sent to TCP that the user is ready for a connection; THP is waiting for notification that a connection has been established (Open Complete event); data received from the user will be discarded and the user notified

8. closed - there is no established connection, nor has one been requested, but the hardware link associated with the user is still active

THP STATE DIAGRAM



APPENDIX D

THP OUTPUT MESSAGES

This appendix shows examples of THP output messages. These messages are sent to the local user to inform him of various situations regarding his input data stream or his connection.

Message texts included in this appendix are meant to show the message style and type of information which will be sent to the user and are not necessarily the exact format to be used for any specific THP implementation. Actual text will be in capital letters. Lower case letters represent dynamic information, such as, subscriber address or user's input command sequence.

Included with each message is a brief description of the circumstances under which the message would be sent to the user.

It should be noted that each message, when sent to the user, will be preceded by the user's prefix character. Also, a user may select, via the set command the message format to be sent to him. He may select one of the following:

1. verbose - entire message as shown, preceded by user's prefix character

2. concise - only message number, preceded by user's prefix character

3. none - no THP output messages will be sent to the user

THP OUTPUT MESSAGES

<01> COMMAND ALREADY IN EFFECT (input command sequence)

The user has entered a command sequence that requests a mode or state that is already in effect, such as: "time on" when the connection inactivity timer is active, "transparent on" when transparent mode is the current data scanning mode, or "mode record" when the current packaging mode is record.

<02> INVALID COMMAND (input command sequence)

The user has entered a command sequence that is not recognized as a valid THP command. The prefix character was not followed by a defined connection control command, NVT control command, or data transmission control character.

<03> INVALID COMMAND FOR CURRENT MODE OR STATE (input command sequence)

The user has entered a command sequence that contains a valid THP command but one that is not allowed during the current mode or state. Examples of this situation are: an abort command when a connection has not been established or requested, a packet release command in binary mode, and an echo command when the RCTE option is in effect.

<04> INVALID SUBCOMMAND PARAMETER (input command sequence)

The user has entered a command sequence that contains a valid THP command but with invalid or incomplete parameters, e.g., a packet release command sequence with no mechanism specified.

<05> COMMAND TERMINATOR NOT FOUND (input command sequence)

The user has entered a command sequence that did not contain the terminating period.

<06> INVALID OPTION PARAMETER input command sequence

The user has entered a valid option command but with invalid or incomplete parameters, e.g., an option command to start a line width option without the line width indicator (value).

<07> INVALID OPTION TYPE input command sequence

The user has entered a valid option command but with an invalid option type.

<08> CONNECTION CLOSING DUE TO INCOMPATIBLE CROSS CONNECTION

The connection is not allowed according to the approved cross-connection matrix. Either the local user's type cannot open a connection to the remote connection user's type or the reverse.

<09> OPTION PENDING input command sequence

The user has entered a valid option command sequence duplicating a request that is currently being negotiated with the remote THP.

<10> OPTION CURRENTLY IN EFFECT input command sequence

The user has entered a valid option command sequence requesting that an option be started that is currently in effect.

<11> OPTION CURRENTLY NOT IN EFFECT input command sequence

The user has entered a valid option command sequence requesting that an option be stopped that was not previously started.

<12> OPTION NOT NEGOTIABLE input command sequence

The user has entered a valid option command sequence but one requesting an option that is not negotiable, according to the user's profile. The negotiability is set up during system generation and is changeable, via the set command, for most of the options.

<13> OPTION QUEUE FULL

The user has entered a valid option command sequence, however, there are already six option requests pending. Six requests, considered to be a reasonable limit, fills THP's option queue. The user must wait until one of the six pending requests is answered by the remote THP.

<14> OPTION REQUEST FOR operator,type ACCEPTED

The user's previously entered option request has been put into effect. The entire request (user's input command sequence) will not be repeated, however, the operator (option command) and option type will be included in the acceptance notification.

<15> OPTION REQUEST FOR operator,type DENIED

The user's previously entered option request has been rejected by the remote THP. The entire request (user's input command sequence) will not be repeated, however, the operator (option command) and option type will be included in the rejection notification. No reason is given for the rejection.

<16> RECEIVING IN BINARY MODE

This message is sent to the user when the characteristics option processing during opening has been completed. The results of the option negotiation were such that the user's receive side can be placed in "automatic" binary mode; the user's characteristics matched exactly, requiring no NVT conversions.

<17> RECEIVING IN NVT MODE

This message is sent to the user when the characteristics option processing during opening has been completed. The results of the option negotiation were such that the user's receive side could not be placed in "automatic" binary mode. That side of the connection remains in normal NVT mode.

<18> TRANSMITTING IN BINARY MODE

This message is sent to the user when the characteristics option processing during opening has been completed. The results of the option negotiation were such that the user's send side could be placed in "automatic" binary mode, requiring no NVT conversions.

<19> TRANSMITTING IN NVT MODE

This message is sent to the user when the characteristics option processing during opening has been completed. The results of the option negotiation were such that the user's send side could not be placed in "automatic" binary mode. That side of the connection remains in normal NVT mode.

<20> OPTION TYPE NOT SUPPORTED input command sequence

The user has entered an option command sequence containing a valid option type (one that is defined by AUTODIN II CCJ/TAC THP), but the option type has not been implemented in the user's supporting THP.

<21> OPTION OPERATOR NOT SUPPORTED input command sequence

The user has entered an option command sequence containing the DOS option operator. CCJ/TAC THPs do not support the DOS option operator (see Paragraph 5.2 of the THP Specification).

<23> CONNECTION CLOSED TO destination subscriber address

This message is sent to the user when a connection closing is completed. A reason may be included with this message, such as: user request, closed by remote user, or protocol error.

<24> CONNECTION CLOSING

This message is sent to the user in response to a status command.

<25> OPENING CONNECTION STATE

This message is sent to the user in response to a status command when the connection state is "opening."

<26> CONNECTION ESTABLISHED TO destination subscriber address

This message is sent to the user upon completion of the open processing for his requested connection or in response to a status command when the connection is active and no hold offs are in effect.

<27> CONNECTION ABORTING

This message is sent to the user upon recognition of an abort condition, such as: TCP closing or THP protocol error.

<28> THP HOLD OFF IN EFFECT

This message is sent to the user in response to a status command. THP will "hold off" user-to-network data during option negotiation, as required, and because of CCJ/TAC flow control mechanisms.

<30> CONNECTION DOES NOT EXIST

This message is sent to the user in response to a status command when there is not an active connection.

<31> TCP TRANSMIT HOLD OFF

This message is sent to the user in response to a status command when there exists a TCP hold off on the requester's send side of the active connection.

<32> TCP RECEIVE HOLD OFF

This message is sent to the user in response to a status command when there exists a TCP hold off on the requester's receive side of the active connection.

<33> COMMAND NOT SUPPORTED input command sequence

The user has entered a command which is not supported on this CCU/TAC for this particular user. The command may have been administratively restricted from use by this user.

<34> INVALID LOCAL PORT-ID

The user has entered an open, listen, or move command containing a source port ID field that is formatted correctly but with a user ID that is not valid for this CCU.

<35> RESOURCES DO NOT CURRENTLY EXIST FOR CONNECTION

The user has entered an open or listen request and the connection is being denied by TCP on the basis of low resources in the CCU.

<37> INVALID OPEN SUBSCRIBER ERROR input command sequence

The user has entered an open command containing an invalid subscriber address specification.

<38> INVALID LISTEN SUBSCRIBER ERROR input command sequence

The user has entered a listen command containing an invalid subscriber address specification.

<39> INVALID OPEN PORT-ID ERROR input command sequence

The user has entered an open command containing an invalid port ID specification.

<40> INVALID LISTEN PORT-ID ERROR input command sequence

The user has entered an listen command containing an invalid port ID specification.

<41> INVALID OPEN P/S/T ERROR input command sequence

The user has entered a open command containing an invalid precedence, security, or TCC specification.

<42> INVALID LISTEN P/S/T ERROR input command sequence

The user has entered a listen command containing an invalid precedence, security, or TCC specification.

<43> OPTION operator, type AGREED TO

This message is sent to the user whenever an unsolicited request to start or stop the binary, go-ahead, RCTE, or XASCII option has been agreed to.

<44> CONNECTION PREEMPTED

This message is sent to the user to inform him that the current connection is being preempted and that a new connection will be established. The user will receive the "connection established" notice some time after the receipt of this message.

<45> ILLEGAL SECURITY

The user has entered an open or listen request specifying a security that is not authorized for his use.

<46> ILLEGAL PRECEDENCE

The user has entered an open or listen request specifying a precedence that is not authorized for his use.

<47> ILLEGAL TCC

The user has entered an open or listen request specifying a TCC that is not authorized for his use.

<49> CONNECTION NOT ESTABLISHED - INPUT DATA DISCARDED

The user has entered data (with or without valid commands) when his connection is in closed or listening state. The message is sent to ensure that the user is aware of the situation.

<51> LISTENING CONNECTION STATE

This message is sent to the user in response to a status command when the connection state is "listening."

<52> MOVE CONNECTION REQUEST DENIED

This message is sent to a user when a move connection request is denied, i.e., designated port ID is unavailable or not authorized to use the connection (security, precedence, or TCC restrictions).

APPENDIX E

THP MCCU DATA STRUCTURES

This appendix contains four major THP data structures of the MCCU:

- Connection Data Structure (CDS)
- Connection Management Block (CMB)
- User's Profile Block (UPB)
- Network Virtual Terminal Table (NVTTAB)

The exact format for each structure, as shown here, is unique to one implementation of the MCCU. Similar information concerning the user's characteristics and connection would most likely be maintained by any THP. These structures provide an example of the type of information needed; they are implementation-oriented by nature.

The format, when shown, is in standard DEC PDP-11 memory image format, with the least significant byte on the right (even number) and the most significant byte on the left (odd number) of a 16-bit word. The bits are labeled 0 to 15 (least significant to most significant) right to left in the word.

The memory image for each structure shown in this appendix is described by an accompanying narrative. The size (byte, word, etc.) of each unit within the structure may cause "extra" bytes within or at the end of the structure. These extra bytes are not used and may be marked as such or not defined in the narrative. They will be represented by a series of dashes (- - -) in the memory image and are always zero. If the extra byte(s) is within the structure, it is, of course, counted in the length of the structure. If the extra byte(s) is at the end of the structure, it will not be counted in the length of the structure.

The description for each structure includes a definition of each field, including the size of the field. Formats (memory images) are included for the CMB and UPB only, as the CDS is primarily made up of these structures.

CONNECTION DATA STRUCTURE (CDS)

The CDS contains all the data structures required by an MCCU TMP to perform connection and data stream processing. There is one CDS per user. The CDS contains the CMB, UPB, and NVTTAB.

The memory image is not shown for the CDS as it is merely the CMB, followed by the UPB, followed by the NVTTAB.

MNEMONIC	# OF BITS	DESCRIPTION
CCUCMB	336	Connection Management Block (CMB) as defined for CCU in Data Structure Specification C-B for MCCU.
CCUUPB	1712	User Profile Block (UPB) as defined for CCU in Data Structure Specification PB for MCCU.
NVTTAB	144[illegible]	Network Virtual Terminal Conversion Table (NVTTAB) as defined in Data Structure Specification for NVTTAB for MCCU.

CONNECTION MANAGEMENT BLOCK (CMB)

This CMB is the MCCU THP data structure which contains connection-oriented information for a user. The CMB is part of the CDS, and, as such, there is one per user.

CONNECTION MANAGEMENT BLOCK (CMB)
11-FEB-79

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	CLCN		0
3	CSURST	CSTATE	2
5	CPRC	CSEC	4
7	CTCC		6
9	CFSUB		8
11	CFPORT		10
13	CTNAID		12
15	CTNNXT		14
17	CTUAID		16
19	CTUNXT		18
21	CLSTRM		20
23	CCMORF		22
25	CPROCS		24

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
27	CRCVQ		26
29	CSNDQ		28
31			30
33	CSNDCW	CSNDWN	32
35	CTUCWN	CTNHLD	34
37	CMVPRT		36
39	COPNEG		38
41	CTMRID		40

CONNECTION MANAGEMENT BLOCK (CMB)
30-JUN-79

MNEMONIC	# OF BITS	DESCRIPTION
CLCN	16	Local Connection Name. This is returned by TCP in the open return event. TCP uses it to uniquely identify the connection.
CSTATE	8	Connection State. 0 = disconnected; 1 = closed; 2 = opening; 3 = listening; 4 = active; 5 = closing; 6 = aborting; 7 = disconnecting.
CSUBST	8	Connection Substate. 1 = wait for characteristics record; 2 = wait for Close Return event; 4 = wait for option negotiation; 8 = wait for TCP window to open; 16 = wait to move connection; 32 = flush until Data Mark record received; 64 = wait for HSI window to open. (More than one condition may exist at any one time).
CSEC	8	Security Code. This field contains the security code associated with an active connection. The value is taken from the Open Complete event from TCP.
CPRC	8	Precedence Level. This field contains the precedence level of an active connetion. The value is taken from the Open Complete event from TCP and is used in all events going to HSI.
CTCC	16	TCC value. This field contains the binary value for the TCC of an active connection. It is acquired by table look-up using the alphabetic triad entered by the user in his open or listen command sequence, if given, or is returned by TCP in the Open Complete event.
CFSUB	16	Foreign Subscriber Address. This field contains the foreign subscriber address for an active connection. The value is returned by TCP in the Open Complete event.
CFPORT	16	Foreign Port ID. This field contains the foreign port ID, as applicable, for an active connection. The value is returned by TCP in the Open Complete event.

MNEMONIC	# OF BITS	DESCRIPTION
CTNBID	16	Current Tonnet Buffer ID. This is the buffer tag returned by the BEC when the buffer is initially requested via a GETBUF call.
CTNNXT	16	Offset to Next Byte in Tonnet Buffer. This field is actually the index to the byte array to which the buffer is mapped. It is always pointing to the position after the last byte written.
CTUBID	16	Current To-user Buffer ID. This is the buffer tag returned by the BEC when the to-user output buffer is initially requested.
CTUNXT	16	Offset to Next Byte of To-user Buffer. This field is an index into the byte array to which the to-user buffer is mapped. It points to the position after the last byte written.
CLSTRM	16	Offset to Last Record Mark. This field is set whenever a record header is written to the tonnet buffer. It is used to calculate and insert the record length when the record is completed.
CCMDBF	16	Current Command Buffer ID. This field is set during command processing, and contains the tag of the buffer used to accumulate the command string from the user's data stream. The field is cleared when the buffer is released after processing the last command in a command sequence.
CPROCQ	16	Current Process Queue Buffer ID. This is the buffer tag used to map to the process queue. If there is no current process queue, the field is zero.
CRCVQ	16	Current Receive Queue Buffer ID. This is the buffer tag used to map to the receive queue. If there is no current receive queue, the field is zero.
CSNDQ	32	Send Queue. Contains tonnet buffer tag and byte count of tonnet buffer waiting for TCP window to open. Only one such tonnet buffer will exist.

MNEMONIC	# OF BITS	DESCRIPTION
CSNDWN	8	Maximum TCP Send Window Size. Number of Send events which may be outstanding to TCP at any given time. Initial value obtained from Open Complete event.
CSNDCW	8	Current TCP Send Window. Number of Send events which may still be sent to TCP.
CTNHLD	8	Connection Flags Byte. 1 = send queue and telnet buffer both exist; 2 = omit auto-listen function on close return (for command override purposes).
CTUCWN	8	Current MSI Window. Number of To User events which may still be sent to MSI.
CMVPRT	16	Move-To Port ID. Binary value of port ID to receive this connection if TCP authorizes the move.
COPNEG	16	Option Negotiation Queue Buffer ID. This field contains the buffer tag used to map to the option negotiation queue which is used when negotiations have been initiated by this THP. If option negotiation is not pending, the field is zero.
CTMRID	16	Time Out Timer ID. This is the timer ID returned by the BEC in a SETMR call which refers to the THP time out event. When an event is processed, this is the timer ID used to reset the timer.

USER PROFILE BLOCK (UPB)

This UPB is the MCCU THP data structure which contains all the information required to perform processing on the data coming from or going to the user. Essentially, the UPB contains the characteristics of the user, both send and receive. The UPB is part of the CDS, and, as such, there is one per user.

USER PROFILE BLOCK (UPB)
11-FEB-79

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	- - -	PPATH	0
3	PSPEED		2
5	PTIMER		4
7	PPACSZ		6
9	PRECSZ		
11	PPRC	PSCRTY	10
13	PTCC		12
15	PRADDR		14
17	PRPORT		
19	PLPORT		18
21	PCMDAD		20
23	PBFSIZ		22
25	PCHRCT		24

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
27	*AB	*AA	MSGFLG		TXTREL		PCKREL		PTYPE								26
29	*AP	*AO	*AN	*AM	AUTOTY		*AL	*AK	*AJ	*AI	*AH	*AG	*AF	*AE	*AD	*AC	28
31	PSDHT																30
33																	32
35																	34
37	PSHTAB																36
39																	38
41																	40
43	PSHHT																42
45																	44
47																	46
49	PUHTAB																48
51																	50

BYTE	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	BYTE
53		52
55		54
57	PSDVT	56
59		58
61		60
63	PSVTAB	62
65		64
67		66
69	PSHVT	68
71		70
73		72
75	PUVTAB	74
77		76

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
79	PSPAGS	PSDYPS	[illegible]
81	PUPAGS	PSPAPL	[illegible]
83	PSLNWD	PSDYL[illegible]	[illegible]
85	PULN[illegible]D	PSPAP[illegible]	[illegible]
87	PSLCNT	PSCCNT	[illegible]
89	PSCROD	PSCROP	[illegible]
91	PSLFOD	PSLFOP	[illegible]
93	PSFFOD	PSFFOP	92
95	PSHTOD	PSHTOP	[illegible]
97	PSVTOD	PSVTOP	[illegible]
99	PNULCT	PNUL[illegible]X	[illegible]
101	PUS[illegible]	PUPC	[illegible]
103	PUCR	PUNUL	[illegible]

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
105	PUFF	PULF	104
107	PUVT	PUHT	106
109	PUEL	PUEC	108
111	PUGA	PUAYT	110
113	PUSI	PUSC	112
115	PURS	PUREL	114
117	PUIF2	PUIF1	116
119	PUIF4	PUIF3	118
121	PUIF6	PUIF5	120
123	PUIF8	PUIF7	122
125	PUIF10	PUIF9	124
127	PUIF12	PUIF11	126
129	PUIF14	PUIF13	128

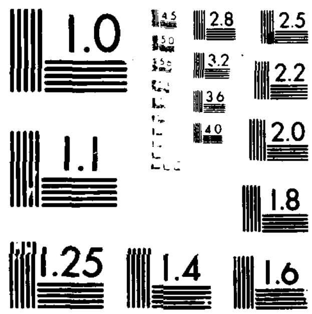
BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
131	PUIF16	PUIF15	130
133	PUIF18	PUIF17	132
135	PSNULL	PUIF19	134
137	PSLF	PSCR	136
139	PSHT	PSFF	138
141	PSEC	PSVT	140
143	PSAVT	PSEL	142
145	PSSO	PSGA	144
147	PSAEL	PSSI	146
149	PSIF1	PSPS	148
151	PSIF4	PSIF3	150
153	PSIF7	PSIF6	152
155	PSIF10	PSIF9	154

AD-A081 757 WESTERN UNION TELEGRAPH CO MCLEAN VA GOVERNMENT SYST--ETC F/G 17/2
COMPUTER PROGRAM DEVELOPMENT SPECIFICATION TERMINAL-TO-HOST PRO--ETC(U)
FEB 80 DCA200-C-637

UNCLASSIFIED SBIE-AD-E100 340 NL

3 OF 4

AD
A081757



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

USER PROFILE BLOCK (UPB)
11-FEB-79

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
157	PSIF13								PSIF11								156
159	PSIF16								PSIF14								158
161	PSIF19								PSIF17								160
163																	162
165																	164
167																	166
169																	168
171																	170
173																	172
175																	174
177																	176
179																	178
181																	180

BYTE	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	BYTE
183		182
185		184
187		186
189	PORTAB	188
191		190
193		192
195		194
197		196
199		198
201		200
203		202
205		204
207		206

USER PROFILE BLOCK (UPB)
11-FEB-79


```
BYTE  15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0   BYTE
     +                                                                +
     !                                                                !
209  !                                                                !   208
     !                                                                !
     +                                                                +
     !                                                                !
211  !                                                                !   210
     !                                                                !
     +                                                                +
     !                                                                !
213  !                                                                !   212
     !                                                                !
     +----------------------------------------------------------------+
```

USER PROFILE BLOCK (UPB)
30-JUN-79

MNEMONIC	# OF BITS	DESCRIPTION
PPATH	8	Path ID. This identifies the host channel with which this UPB is associated. It is used in the HSI-THP interface.
PSPEED	16	Line Speed. Code representing speed of input (host-CCU) line, needed by TCP in open request for flow control algorithm: (in bps) 0 = 150 or less; 1 = 300; 2 = 600; 3 = 1200; 4 = 2400; 5 = 4800; 6 = 9600; 7 = 19200 or more.
PTIMER	16	Time Out Value. This field contains the value to be used in the connection inactivity timer for the Connection Timeout event. The value is in tenths of seconds and has a range of 0-32767.
PPACSZ	16	Packet Size. This field contains the maximum number of bytes in a packet. The value is the same as the maximum segment size unless the packet release mechanism is set to release on specific number of characters.
PRECSZ	16	Record Size. This field contains the maximum number of characters for transmission to the user. It is used when the text release mode is set for release of specific number of characters.
PSCRTY	8	Security Code. Contains value of security code to be used for auto-open and auto-listen processing in building open request to TCP.
PPRC	8	Precedence Level. Contains value of precedence to be used for auto-open and auto-listen processing in building open request to TCP.
PTCC	16	TCC Code. Contains the binary value representing the TCC to be used for auto-open and auto-listen processing in building Open event to TCP.
PRADDR	16	Remote Subscriber Address. Contains remote subscriber address to be used for auto-open and auto-listen processing in building open request to

MNEMONIC	# OF BITS	DESCRIPTION
		TCP.
PRPORT	16	Remote Port ID. Contains remote port ID, if applicable, to be used for auto-open and auto-listen processing in building open request to TCP.
PLPORT	16	Local Port ID. Contains binary value for port ID for local user. Must be specified if user is auto-open/listen type or a user which will be receiving a "moved" connection (i.e., the user to be specified in a move command sequence).
PCMDWD	16	Command Word. Indicates which TMP commands are valid for user to input; one bit per command (or command type); bit is set if user may input corresponding command; offsets from B=15 are for: open, listen, close, abort, status, set, packet release, transparent, mode, move, RCTE, options (DOR, DNR, etc.), echos (echo, full and half duplex), and time commands.
PBFSIZ	16	Buffer Size. Value to be used for buffer size in GETBUF calls, based on "average letter size" of this user.
PCHRCT	16	Input Character Count. Value representing current number of characters which remain to be input by user before packet release on number of characters occurs.
PTYPE	8	Cross-Connection Matrix Type. Contains value of user type to be used in determining cross-connection compatibility during open characteristics checking.
PCKREL	2	Packet Release Mechanism. 00 = full segment or Send Now character (default); 01 = as received from TM/MSI; 10 = detection of new line sequence; 11 = on accumulation of specific number of characters (PPKCSZ).
TXTREL	2	Text Release Mode. Release of To User buffers. 00 = end-of-letter; 01 = one character at a time; 10 = detection of

MNEMONIC	# OF BITS	DESCRIPTION
		new line sequence; 11 = accumulation of specific number of characters (PRECSZ).
MSGFLG	2	Diagnostic Message Flag. 00 = none sent; 01 = abbreviated messages sent; 10 = complete messages sent.
*AA = USRMOD	1	User Mode Flag. Indicates current packaging mode for user-to-network data; 0 = record mode; 1 = stream mode.
*AB = SRVMOD	1	Server Mode Flag. Indicates current mode of network-to-user data; 0 = record mode; 1 = stream mode.
*AC = USRXA	1	User XASCII Flag (currently sending to network). 0 = not in XASCII; 1 = in XASCII mode.
*AD = SRVXA	1	Server XASCII Flag (currently receiving from network). 0 = not in XASCII; 1 = in XASCII mode.
*AE = USRTRN	1	User Transparent Mode Flag. 0 = transparent mode off; 1 = transparent mode on.
*AF = UNLTYP	1	User New Line Type. 0 = CR/LF is new line sequence; 1 = CR only is new line sequence.
*AG = SNLTYP	1	Server New Line Type. 0 = CR/LF is new line sequence; 1 = CR only is new line sequence.
*AH = PCFLG	1	Prefix Continuation Flag. When set this indicates that a prefix character was encountered as last character in buffer. FRMUSR will call PCUSR for completion of command processing when the next From User event is received.
*AI = CRFLG	1	Carriage Return Flag. Used by receive NVT processing in handling new line sequence from network, to ensure that CR/LF, CR only, and LF only conditions are handled properly.
*AJ = RMFLG	1	Record Mark Flag. Used by send NVT processing (DATUSR, et al) in determining requirements for new record header and completion (computing

MNEMONIC	# OF BITS	DESCRIPTION
		length) of previous record.
*AK = TIMFLG	1	Timer Flag, when set inactivity timer is on; if clear user has turned timer off via time command.
*AL = PREFLG	1	Preemptability Flag. Set if user's connections may be preempted by TCP; clear if preemption not allowed; indicator passed to TCP in Open event.
AUTOTY	2	Automatic Feature Type, 0 = non-auto; 1 = auto-listen; 2 = auto-open = (both auto-listen and auto-open may be set)
*AM = SECFLG	1	Security Flag, when set security parameter should be specified for auto-listen in Open event to TCP.
*AN = TCCFLG	1	TCC Flag, when set TCC parameter should be specified for auto-listen in Open event to TCP.
*AO = PRCFLG	1	Precedence Flag, when set precedence parameter should be specified for auto-listen in Open event to TCP.
*AP = ADRFLG	1	Remote Address Flag, when set remote address should be specified for auto-listen in Open event to TCP.
PSDHT	48	Server Dynamic Horizontal Tabs. Contains tab settings in same format as PSHTAB to be used in processing horizontal tab function from network. Designates settings to be used, as requested and agreed upon via option negotiation.
PSHTAB	48	Server Default Horizontal Tabs. Contains tab settings to be used to restore PSDHT to default value when option is terminated. Field is changeable via Set command by user and contains up to six tab stops or 255 in first byte followed by integer representing first tab and incremental value to subsequent tabs.
PSHHT	48	Server Hardware Horizontal Tabs. Contains hardware tab settings in same format as PSHTAB. Used to determine if

USER PROFILE BLOCK (UPB)
30-JUN-79

MNEMONIC	# OF BITS	DESCRIPTION
		simulation is required in performing horizontal tab function on data from network. Value is changeable via Set command by user.
PUHTAB	48	User Horizontal Tab Stops. Contains tab settings in same format as PSHTAB to be used in characteristics compatibility checking during open processing. Value represents way in which user will format his data.
PSDVT	48	Server Dynamic Vertical Tabs. Contains tab settings in same format as PSVTAB to be used in simulation of vertical tab function from network. Field may be set by option negotiation or at SYSGEN time.
PSVTAB	48	Server Default Vertical Tabs. Contains tab settings to be used to restore PSDVT to default value when option is terminated. Field is changeable via Set command by user and contains up to six tab stops or 255 in first byte followed by integer representing first tab and incremental value to subsequent tabs.
PSHVT	48	Server Hardware Vertical Tabs. Contains hardware tab settings in same format as PSVTAB. Used to determine if simulation is required in performing vertical tab function on data from network. Value is changeable via Set command by user.
PUVTAB	48	User Vertical Tab Stops. Contains tab settings in same format as PSVTAB to be used in characteristics compatibility checking during open processing. Value represents way in which user will format his data.
PSDYPS	8	Server Dynamic Page Size. Contains number of lines to be used per page on receive data path. Set as a result of option negotiation.
PSPAGS	8	Server Default Page Size. Contains number of lines to be used per page on receive data path. Value used to restore default value into PSDYPS upon return to default condition via option

MNEMONIC	# OF BITS	DESCRIPTION
		negotiation, Value is changeable via Set command by user.
PSPAPL	8	Server Hardware Paper Length, Contains number of lines restricted by hardware on a page, Used in determining feasibility of option request for page size or vertical tab settings, Value is changeable via Set command by user.
PUPAGS	8	User Page Size, Contains number of lines which will be formated per page by user in his data going toward network, Used in characteristics processing during connection opening.
PSDYLW	8	Server Dynamic Line Width, Contains number of characters per line which have been negotiated to be used on receive data path, Set as result of option negotiation.
PSLNWD	8	Server Default Line Width, Contains number of characters per line to be used on receive data path, Value used to restore default value into PSDYLW upon return to default via option negotiation, Value is changeable via Set command by user.
PSPAPW	8	Server Hardware Paper Width, Contains number of characters restricted by hardware per line, Used in determining feasibility of option request for line width or horizontal tab settings, Value is changeable via Set command by user.
PULNWD	8	User Line Width, Contains number of characters per line which will be formated by user in his data to network, Used in characteristics processing during connection opening.
PSCCNT	8	Server Column Count, Current column count for server line width processing.
PSLCNT	8	Server Line Count, Current line count for server page size processing.
PSCRDP	8	Server Carriage Return (CR) Disposition, Value describing disposition of CR in network-to-user

MNEMONIC	# OF BITS	DESCRIPTION
		data: 0 => 250 = number of nulls to be inserted for timing delay; 254 = discard CR; 255 = deliver as is.
PSCRDD	8	Server CR Default Disposition. Original and unchangeable default disposition value in same format as and used to restore PSCRDP upon return to default condition via option negotiation.
PSLFDP	8	Server Linefeed (LF) Disposition. Value describing disposition of LF in network-to-user data: 0 => 252 = number of nulls to be inserted for time delay; 253 = simulate LF; 254 = discard LF; 255 = deliver as is.
PSLFDD	8	Server LF Default Disposition. Original and unchangeable default disposition value in same format as and used to restore PSLFDP upon return to default condition via option negotiation.
PSFFDP	8	Server Formfeed (FF) Disposition. Value describing disposition of FF in network-to-user data: 0 => 251 = number of nulls to be inserted for time delay; 252 = replace NVT FF with local new line sequence; 253 = simulate FF; 254 = discard FF; 255 = deliver as is.
PSFFDD	8	Server FF Default Disposition. Original and unchangeable default disposition value in same format as and used to restore PSFFDP upon return to default condition via option negotiation.
PSHTDP	8	Server Horizontal Tab (HT) Disposition. Value describing disposition of HT in network-to-user data: 0 => 252 = number of nulls to be inserted for time delay; 253 = simulate HT; 254 = discard HT; 255 = deliver as is.
PSHTDD	8	Server HT Default Disposition. Original and unchangeable default disposition value in same format as and used to restore PSHTDP upon return to default condition via option negotiation.
PSVTDP	8	Server Vertical Tab (VT) Disposition. Value describing disposition of VT in

MNEMONIC	# OF BITS	DESCRIPTION
		network-to-user data: 0 => 252 = number of nulls to be inserted for time delay; 252 = replace NVT VT with local new-line sequence; 253 = simulate VT; 254 = discard FF; 255 = deliver as is.
PSVTDD	8	Server VT Default Disposition. Original and unchangeable default disposition value in same format as and used to restore PSVTDP upon return to default condition via option negotiation.
PNULMX	8	Maximum Null Count. Maximum number of hardware nulls coming from user following CR, LF, FF, VT, or HT for timing delays, up to this number will be stripped when converting from user to NVT value.
PNULCT	8	Current Null Count. Number of nulls remaining to be stripped following CR, LF, FF, VT, or HT. Value is non-zero only when sequence of the character and its trailing nulls is split between two from-user buffers.
PUPC	8	User Prefix Character (PC). Value of PC to be sent by user.
PUSN	8	User Send Now (SN) Character. Value of SN to be sent by user.
PUNUL	8	User Null (NUL) Character. Value of NUL to be sent by user.
PUCR	8	User Carriage Return (CR) Character. Value of CR to be sent by user.
PULF	8	User Linefeed (LF) Character. Value of LF to be sent by user.
PUFF	8	User Formfeed (FF) Character. Value of FF to be sent by user.
PUHT	8	User Horizontal Tab (HT) Character. Value of HT to be sent by user.
PUVT	8	User Vertical Tab (VT) Character. Value of VT to be sent by user.
PUEC	8	User Erase Character (EC) Character. Value of EC to be sent by user.

MNEMONIC	# OF BITS	DESCRIPTION
PUEL	8	User Erase Line Character (EL) Character, Value of EL to be sent by user.
PUAYT	8	User Are-You-There? (AYT) Character, Value of AYT to be sent by user.
PUGA	8	User Go-Ahead (GA) Character, Value of GA to be sent by user.
PUSO	8	User XASCII Shift-out (SO) Character, Value of SO to be sent by user.
PUSI	8	User XASCII Shift-in (SI) Character, Value of SI to be sent by user.
PUBEL	8	User Bell (BEL) Character, Value of BEL to be sent by user.
PUBS	8	User Backspace (BS) Character, Value of BS to be sent by user.
PUIF1	8	User Interrupt Function 1 (IF1) Character, Value of IF1 to be sent by user.
PUIF2	8	User Interrupt Function 2 (IF2) Character, Value of IF2 to be sent by user.
PUIF3	8	User Interrupt Function 3 (IF3) Character, Value of IF3 to be sent by user.
PUIF4	8	User Interrupt Function 4 (IF4) Character, Value of IF4 to be sent by user.
PUIF5	8	User Interrupt Function 5 (IF5) Character, Value of IF5 to be sent by user.
PUIF6	8	User Interrupt Function 6 (IF6) Character, Value of IF6 to be sent by user.
PUIF7	8	User Interrupt Function 7 (IF7) Character, Value of IF7 to be sent by user.
PUIF8	8	User Interrupt Function 8 (IF8)

MNEMONIC	# OF BITS	DESCRIPTION
		Character, Value of IF8 to be sent by user.
PUIF9	8	User Interrupt Function 9 (IF9) Character, value of IF9 to be sent by user.
PUIF10	8	User Interrupt Function 10 (IF10) Character, Value of IF10 to be sent by user.
PUIF11	8	User Interrupt Function 11 (IF11) Character, value of IF11 to be sent by user.
PUIF12	8	User Interrupt Function 12 (IF12) Character, Value of IF12 to be sent by user.
PUIF13	8	User Interrupt Function 13 (IF13) Character, Value of IF13 to be sent by user.
PUIF14	8	User Interrupt Function 14 (IF14) Character, Value of IF14 to be sent by user.
PUIF15	8	User Interrupt Function 15 (IF15) Character, Value of IF15 to be sent by user.
PUIF16	8	User Interrupt Function 16 (IF16) Character, Value of IF16 to be sent by user.
PUIF17	8	User Interrupt Function 17 (IF17) Character, Value of IF17 to be sent by user.
PUIF18	8	User Interrupt Function 18 (IF18) Character, Value of IF18 to be sent by user.
PUIF19	8	User Interrupt Function 19 (IF19) Character, value of IF19 to be sent by user.
PSNULL	8	Server Null (NUL) Character, Value of NUL to be received by user.
PSCR	8	Server Carriage Return (CR) Character, Value of CR to be received by user.

MNEMONIC	# OF BITS	DESCRIPTION
PSLF	8	Server Linefeed (LF) Character, value of LF to be received by user.
PSFF	8	Server Formfeed (FF) Character, value of FF to be received by user.
PSHT	8	Server Horizontal Tab (HT) Character, Value of HT to be received by user.
PSVT	8	Server Vertical Tab (VT) Character, Value of VT to be received by user.
PSEC	8	Server Erase Character (EC) Character, Value of EC to be received by user.
PSEL	8	Server Erase Line (EL) Character, value of EL to be received by user.
PSAYT	8	Server Are-You-There? (AYT) Character, value of AYT to be received by user.
PSGA	8	Server Go-Ahead (GA) Character, value of GA to be received by user.
PSSO	8	Server XASCII Shift-out (SO) Character, Value of SO to be received by user.
PSSI	8	Server XASCII Shift-in (SI) Character, Value of SI to be received by user.
PSBEL	8	Server XBell (BEL) Character, Value of BEL to be received by user.
PSBS	8	Server Backspace (BS) Character, value of BS to be received by user.
PSIF1	8	Server Interrupt Function 1 (IF1) Character, Value of IF1 to be received by user.
PSIF3	8	Server Interrupt Function 3 (IF3) Character, Value of IF3 to be received by user.
PSIF4	8	Server Interrupt Function 4 (IF4) Character, Value of IF4 to be received by user.
PSIF6	8	Server Interrupt Function 6 (IF6) Character, Value of IF6 to be received by user.

MNEMONIC	# OF BITS	DESCRIPTION
PSIF7	8	Server Interrupt Function 7 (IF7) Character, Value of IF7 to be received by user.
PSIF9	8	Server Interrupt Function 9 (IF9) Character, Value of IF9 to be received by user.
PSIF10	8	Server Interrupt Function 10 (IF10) Character, Value of IF10 to be received by user.
PSIF11	8	Server Interrupt Function 11 (IF11) Character, Value of IF11 to be received by user.
PSIF13	8	Server Interrupt Function 13 (IF13) Character, Value of IF13 to be received by user.
PSIF14	8	Server Interrupt Function 14 (IF14) Character, Value of IF14 to be received by user.
PSIF16	8	Server Interrupt Function 16 (IF16) Character, Value of IF16 to be received by user.
PSIF17	8	Server Interrupt Function 17 (IF17) Character, Value of IF17 to be received by user.
PSIF19	8	Server Interrupt Function 19 (IF19) Character, Value of IF19 to be received by user.
POPTAB	416	Option Table, Used by option negotiation modules. One 4-byte entry for each of 13 options defined: byte 0=byte offset to send value; byte 1=byte offset to receive value; byte 2=length of value field; byte 3=flags: (if set) bit 0=DOS successful; bit 1=DOR successful; bit 2=WLS successful; bit 3=WLR successful; bit 4=send side not negotiable; bit 5=receive side not negotiable

NETWORK VIRTUAL TERMINAL TABLE (NVTTAB)

This NVTTAB is used by the user-to-network data path modules in the MCCU THP to determine the required action for each 7-bit ASCII character being processed from the user. There is one byte representing each possible character value (1-128).

NETWORK VIRTUAL TERMINAL TABLE (NVTTAB) PAGE E-33
11-FEB-79

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1		NVTVAL	0
3			2
5			4
7			6
9			8
11			10
13			12
15			14
17			16
19			18
21			20
23			22
25			24

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
27																	2
29																	2
31																	
33																	32
35																	3
37																	3
39																	3
41																	4
43																	4
45																	4
47																	4
49																	4
51																	5

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
53																	52
55																	54
57																	56
59																	58
61																	60
63																	62
65								cont'd									64
67																	66
69																	68
71																	70
73																	72
75																	74
77																	76

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
79																	78
81																	80
83																	82
85																	84
87																	86
89																	88
91																	90
93																	92
95																	94
97																	96
99																	98
101																	100
103																	102

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
105																	104
107																	106
109																	108
111																	110
113																	112
115																	114
117																	116
119																	118
121																	120
123																	122
125																	124
127																	126
129					. . .												128

MNEMONIC	# OF BITS	DESCRIPTION
NVTVAL	8	NVT value. There are 129 of these bytes, each of which represent the required action for any input character value, including normal ASCII characters and characters defined in the UPR. Possible values and the corresponding actions are: 0 = pass as data; <0 (negative) = NVT record value; 1-19 = special action for interrupt function required;
cont'd	1024	20 = send now character; 21 = prefix character; 22 = XASCII shift-out character; 23 = null character (in stream mode this value will be replaced by the NVT null record type value).

APPENDIX F

THP EVENTS FOR MCCU

This appendix describes the events "received" and "sent" by THP in the MCCU. For the network-to-user processing path, THP receives events from TCP and sends events to HSI. For the user-to-network processing path, THP receives events from HSI and queues events to TCP.

These events are implementation-oriented and, in fact, represent one MCCU implementation. This appendix is not intended to provide documentation for the MCCU events; that is provided under separate cover. It is, rather, intended to suggest the type of information that must be passed between a THP and HSI or TCP in an implementation similar to the MCCU.

The format is shown in standard DEC PDP-11 memory image format, with the least significant byte on the right (even number) and the most significant byte on the left (odd number) of a 16-bit word. The bits are labeled 0 to 15 (least significant to most significant) right to left in the word.

The memory image for each structure shown in this appendix is described by an accompanying narrative. The size (byte, word, etc.) of each unit within the structure may cause "extra" bytes within or at the end of the structure. These extra bytes are not used and may be marked as such or not defined in the narrative. They will be represented by a series of dashes (- - -) in the memory image and are always zero. If the extra byte(s) is within the structure, it is, of course, counted in the length of the structure. If the extra byte(s) is at the end of the structure, it will not be counted in the length of the structure.

EVENT SPECIFICATION

EVENT NAME: ABORT EVENT
MNEMONIC: CABORT

CPCI: MCCU

SENDER: TMP

RECEIVER: HSI

PURPOSE:

This event directs HSI to release all buffers queued to the specified channel.

REASON:

TMP has detected a situation which requires a flush of all data going to the user. Currently this event is sent when TCP has closed the connection for some reason.

COMMENTS:

This event has no effect on input buffers or the buffer currently being output on the channel.

VERSION: 13-SEP-78

ABORT EVENT
13-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	. . .	PATHID	2

ABORT EVENT
13-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Abort event = 5.
TRANID	8	Transaction ID, Sequence number for this event, for coordination of return event, with range 1-255.
PATHID	8	Path ID, Internal HSI-TMP user identification value with range 1-32.

EVENT SPECIFICATION

EVENT NAME: CLOSE EVENT
MNEMONIC: OCLOSE

CPCI: MCCU

SENDER: THP

RECEIVER: TCP

PURPOSE:

This event allows THP to request that TCP close a virtual connection.

REASON:

THP issues a Close event in response to a close command from the user or as a result of a protocol error encountered in processing for the referenced connection.

COMMENTS:

If TCP receives a second consecutive Close event for a connection, i.e., a Close event received following another Close event, but before close processing is complete, TCP will send a Close Return event for the first Close event, and treat the second as a "flushing" close request.

VERSION: 29-SEP-78

CLOSE EVENT
29-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	LCN		2
5	CLSTYP		

CLOSE EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, value for Close event = 67.
TRANID	8	Transaction ID, Sequence number for this event, for coordination of return event, with range 1-255.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.
CLSTYP	16	Type of Close, Indicates what type of close should be performed by TCP; 0 = deferred, all Send events issued prior to Close event will be delivered to destination, if possible; 1 = immediate, all Send events not yet segmentized will be returned and TCP close processing done immediately; in either case, no further Receive events will be queued to THP.

EVENT SPECIFICATION

EVENT NAME: CLOSE RETURN EVENT
MNEMONIC: OCLSRT

CPCI: MCCU

SENDER: TCP

RECEIVER: THP

PURPOSE:

This event notifies THP that TCP close processing has been completed for a connection.

REASON:

The event will be issued in response to a previous Close event or for several exception conditions encountered by TCP, such as, remote user closing, S/P/T errors, destination subscriber down, etc.

COMMENTS:

If TCP sends an unsolicited Close Return event, i.e., one not caused by previous Close event, the Transaction ID of the return event will be zero.

VERSION: 29-SEP-78

CLOSE RETURN EVENT
29-SEP-78

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
1	TRANID								OPCODE								0
3	LCN																2
5	CLRSTA																4

CLOSE RETURN EVENT
02-MAR-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, value for Close Return event = 43.
TRANID	8	Transaction ID. Sequence number for previous CLOSE event, if any, with range 1-255; this byte will be zero if the return event is unsolicited. For a Local Close, this field contains a THP Close reason if a flushing close was sent to TCP.
LCN	16	Local Connection Name. Internal THP-TCP connection identification value with range 1-32.
CLRSTA	16	Close Status. 0=successful local; 1=nonexistent connection; 11=remote close; 16=SCM sec error; 17=SCM prec error; 18=SCM TCC error; 19=SCM addr error; 20= destntn subscriber down; 21=destntn access circuit down; 22=destntn busy; 30= remote TCP sec error; 31=remote TCP TCC error; 32=remote preempt; 33=space preempt; 34=half open; 35=no ack rcvd; 36=line error; 37=SCM down

EVENT SPECIFICATION

EVENT NAME: LINE CONNECT EVENT
MNEMONIC: OCONCT

CPCI: MCCU

SENDER: HSI

RECEIVER: THP

PURPOSE:

This event informs THP that the referenced user has been activated at the link (channel) level, e.g., link protocol has been initialized.

REASON:

THP must be informed when the user is available so that data structures may be initialized and automatic listens may be initiated, as required.

COMMENTS:

For most channels this event will be sent only once, upon the first input received from the host. This input may be actual user text or protocol control characters.

VERSION: 13-SEP-78

LINE CONNECT EVENT
13-SEP-79

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	...	OPCODE	0
3	...	PATHID	2

LINE CONNECT EVENT
13-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Line Connect event = 33.
- - -	8	This byte is not used for this event.
PATHID	8	Path ID, Internal HSI-TMP user identification value with range 1-32.

EVENT SPECIFICATION

EVENT NAME: LINE DISCONNECT EVENT
MNEMONIC: CDISCT

CPCI: MCCU

SENDER: MSI

RECEIVER: THP

PURPOSE:

This event informs THP that a condition has been detected by MSI which indicates that a connection established or requested for this user should be closed immediately.

REASON:

THP must be informed when such a condition occurs so that the connection, if any, can be closed and resources released.

COMMENTS:

The conditions which cause Line Disconnect events to be queued to THP will be host specific, i.e., as specified by the requirements of each host-MCCU configuration.

VERSION: 13-SEP-78

LINE DISCONNECT EVENT
13-SEP-78

BYTE	15–8	7–0	BYTE
1	- - -	OPCODE	0
3	- - -	PATH-ID	2

LINE DISCONNECT EVENT
13-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Line Disconnect event = 32.
- - -	8	This byte is not used for this event.
PATHID	8	Path ID, Internal HSI-THP user identification value with range 1-32.

EVENT SPECIFICATION

EVENT NAME: FROM USER EVENT
MNEMONIC: CF-USR

CPCI: MCCU

SENDER: HSI

RECEIVER: TMP

PURPOSE:

This event passes a buffer of user data or control information to TMP.

REASON:

HSI has received data/control information over the channel associated with this user (path ID). The data/control information must be analyzed by TMP and processed as required by the user's data stream. This includes formatting data for transmission over the network as well as responding to requests for some service (TMP commands).

COMMENTS:

VERSION: 16-SEP-78

FROM USER EVENT
16-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	- - -	PATHID	2
5	BUFID		
7	BYTCNT		6
9	COMDAT		8
11	OFFSET		10

FROM USER EVENT
02-MAR-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for From user event = 35,
TRANID	8	Transaction ID, Sequence number for this event, for coordination of return event, with range 0-255,
PATHID	8	Path ID, Internal HSI-THP user identification value with range 1-32,
BUFID	16	Buffer ID, Tag used to map to the buffer containing user data; zero if From User event carries control information only,
BYTCNT	16	Byte Count, Number of bytes of data in the user buffer, excluding all link protocol characters,
CONDAT	16	Control Data, Information which cannot be conveyed in the user data stream, such as, hardware break signal, Currently only the hardware break signal is identified, If bit 0 = 1, hardware break signal was received,
OFFSET	16	Data Offset, Offset to start of actual data characters in buffer, beyond any link protocol characters, This value is relative to 0,

EVENT SPECIFICATION

EVENT NAME: FROM USER RETURN EVENT
MNEMONIC: OFUSRT

CPCI: MCCU

SENDER: TMP

RECEIVER: HSI

PURPOSE:

This event informs HSI that the data buffer/control information referenced in the corresponding From User event has been processed and the buffer, if any, has been released by TMP.

REASON:

HSI must know when each From User is processed as the number of outstanding (unacknowledged) From User events plays a part in HSI's portion of the MCCU flow control mechanism.

COMMENTS:

VERSION: 01-SEP-78

FROM USER RETURN EVENT
01-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	PRILVL	PATHID	2

FROM USER RETURN EVENT
02-MAR-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for From User Return event = 6.
TRANID	8	Transaction ID, Sequence number of the corresponding From User event with range 0-255.
PATHID	8	Path ID, Internal HSI-THP user identification value with range 1-32.
PRILVL	8	Priority Level, Category of connection, used in considering action required in low buffer situations.

EVENT SPECIFICATION

EVENT NAME: INITIALIZE TMP EVENT
MNEMONIC: CINTMP

CPCI: MCCU

SENDER: REC

RECEIVER: TMP

PURPOSE:

This event informs TMP that the MCCU has been initialized and provides TMP with information necessary to initialize its data structures.

REASON:

TMP must be given access to the Default Profile Blocks and told how many users are on this MCCU so that data structures can be created for each user, upon receipt of each Line Connect event.

COMMENTS:

VERSION: 28-FEB-79

INITIALIZE THP EVENT
08-FEB-79

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
1	NOPATH								OPCODE								0
3	OPBTAG																2

INITIALIZE THP EVENT
28-FEB-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Initialize THP event = 37.
NOPATH	8	Number of Paths, Maximum number of active connections for this MCCU.
DPBTAG	16	Default Profile Block Buffer ID, Tag to be used to map to the buffer containing the default UPBs for this MCCU. The first 32 words contain offsets into the buffer for each UPB, using the PathId (from MSI) as the index into the 32-word area.

EVENT SPECIFICATION

EVENT NAME: INTERRUPT EVENT
MNEMONIC: OINTPT

CPCI: MCCU

SENDER: THP

RECEIVER: TCP

PURPOSE:

This event requests that TCP perform either an interrupt or flush function for data being sent from the source (originator of Interrupt event) to the destination on the referenced connection.

REASON:

The user is given 19 interrupt function "keys" which cause 12 different interrupt functions. Of these one causes an out-of-band TCP interrupt function and several cause TCP flush functions. These TCP interrupt/flush functions are requested via the Interrupt event.

COMMENTS:

VERSION: 29-SEP-78

INTERRUPT EVENT
29-SEP-78

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
1	TRANID								OPCODE								2
3	LCN																2
5	• • •								INTYPE								4

INTERRUPT EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Interrupt event = 69.
TRANID	8	Transaction ID, Sequence number for this event, for coordination of return event, with range 1-255.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.
INTYPE	8	Function Type, Value indicating type of request: 0 = out-of-band interrupt; 1 = flush.

EVENT SPECIFICATION

EVENT NAME: INTERRUPT RETURN EVENT
MNEMONIC: CINTRT

CPCI: MCCJ

SENDER: TCP

RECEIVER: THP

PURPOSE:

This event informs THP that the interrupt/flush function requested in a previous Interrupt event has been completed. If there was not a previous Interrupt event (Transaction ID is zero), this event informs THP that an interrupt/flush function was requested by the remote THP.

REASON:

Source THP has requested an interrupt/flush and both source and destination THPs must be made aware that the function has been performed so that any THP followup processing may be completed.

COMMENTS:

VERSION: 12-OCT-78

INTERRUPT RETURN EVENT
12-OCT-78

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
1	TRANID								OPCODE								0
3	LCN																2
5	INTSTA								INTYPE								4

INTERRUPT RETURN EVENT
12-OCT-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Interrupt Return event = 39.
TRANID	8	Transaction ID, Sequence number for corresponding Interrupt event with range 1-255; zero if this is unsolicited Interrupt Return event.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.
INTYPE	8	Function Type, Value indicating type of function: 0 = out-of-band interrupt; 1 = flush.
INTSTA	8	Return Status, 0 = successful; 1 = connection does not exist; 2 = function not legal for current TCP state; 7 = function not completed (undelivered).

EVENT SPECIFICATION

EVENT NAME: MOVE CONNECTION EVENT
MNEMONIC: OMOVE

CPCI: MCCU

SENDER: TMP

RECEIVER: TCP

PURPOSE:

This event requests that TCP verify a user as authorized to receive an already established connection.

REASON:

TMP has received a request on an active connection, via the move command, that the connection be "moved" to another user on this MCCU. TMP must ensure that the user is authorized the security, precedence, and TCC of the established connection. To acquire this authorization, TMP enlists TCP.

COMMENTS:

VERSION: 29-SEP-78

MOVE CONNECTION EVENT
29-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	LCN		2
5	PORTID		4

MOVE CONNECTION EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Move Connection event = 74.
TRANID	8	Transaction ID, Sequence number of this event, for coordination with return event, with range 1-255.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.
PORTID	16	Local Port ID, Value for local port ID for new user; bits 1-11 = user ID; bits 12-15 = function suffix; bit 0 = static/dynamic port indicator.

EVENT SPECIFICATION

EVENT NAME: MOVE CONNECTION RETURN EVENT
MNEMONIC: OMOVRT

CPCI: MCCU

SENDER: TCP

RECEIVER: THP

PURPOSE:

This event responds to THP's Move Connection event, indicating whether the new user is authorized to receive the established connection.

REASON:

TCP received a Move event from THP requesting that a verification be made as to the authorization of a new user to use an established connection.

COMMENTS:

VERSION: 29-SEP-78

MOVE CONNECTION RETURN EVENT
29-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	LCN		2
5	- - -	MOVSTA	4

MOVE CONNECTION RETURN EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Move Connection Return event = 48.
TRANID	8	Transaction ID, Sequence number of the corresponding Move Connection event with range 1-255.
LCN	16	Local Connection Name, internal THP-TCP connection identification value with range 1-32.
MOVSTA	8	Status of Move event processing: 0 = new user is authorized to receive referenced connection; 1 = new user is not authorized to receive referenced connection.

EVENT SPECIFICATION

EVENT NAME: OPEN COMPLETE EVENT
MNEMONIC: OOPCMP

CPCI: MCCU

SENDER: TCP

RECEIVER: THP

PURPOSE:

This event notifies THP of the security, precedence, TCC, and foreign address of the connection. These fields may not have been specified in the original open event or in the case of precedence may have been changed by TCP.

REASON:

This event is issued after the first "SYN" segment is received from the remote TCP for the connection. At that time all unspecified parameters of the open request become bound by the values of the incoming segment.

COMMENTS:

The local user must be authorized the higher precedence and all values associated with the new connection that had been previously unspecified in his open request.

VERSION: 26-FEB-79

OPEN COMPLETE EVENT
29-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	LCN		2
5	PREC	SEC	4
7	TCC		6
9	FORSUB		8
11	FORPRT		10

OPEN COMPLETE EVENT
29-SEP-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Open Complete event = 45.
TRANID	8	Transaction ID, Sequence number of the corresponding Open event with range 1-255.
LCN	16	Local Connection Name, Internal TMP-TCP connection identification value with range 1-32.
SEC	8	Connection Security Level, 15 is high; 0 is low.
PREC	8	Connection Precedence Level, 15 is high; 0 is low.
TCC	16	Connection TCC, Binary value with range 0-512.
FORSUB	16	Foreign Subscriber ID.
FORPRT	16	Foreign Port ID.

EVENT SPECIFICATION

EVENT NAME: OPEN EVENT
MNEMONIC: OOPEN

CPCI: MCCU

SENDER: TMP

RECEIVER: TCP

PURPOSE:

This event allows TMP to specify the parameters of a new connection.

REASON:

TMP builds an Open event when it receives an open or listen command from the user when a connection does not already exist, or when it performs an auto-listen or auto-open function. In the case of a listen request, the destination address, security, precedence, or TCC may be unspecified.

COMMENTS:

The immediate return event (Open Return event) merely signifies that the parameters specified in the Open event were valid for the user. The connection, however, is not established until TCP completes the three-way handshake. This will be caused by a Send event by either the local or remote TMP.

VERSION: 08-FEB-79

OPEN EVENT
08-FEB-79

BYTE	15–8	7–0	BYTE
1	TRANID	OPCODE	0
3	OPNFLG	PREFLG	2
5	PREC	SEC	4
7	TCC		6
9	FORSUB		8
11	FORPRT		10
13	LOCPRT		12
15	SPO	LETSIZ	14

OPEN EVENT
28-FEB-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for an Open event = 64.
TRANID	8	Transaction ID, Sequence number for this event, for coordination of return events, with range 1-255.
PREFLG	8	Preemption Flag, Falg indicating if this connection can be preempted: 0 = preemption not allowed; 1 = connection can be preempted.
OPNFLG	8	Open Flag Byte, Flags indicating the unspecified parameters of the open request. If set, bit 0 = no security; bit 1 = no precedence; bit 2 = no TCC; bit 3 = no foreign subscriber address.
SEC	8	Security, Value for security level for the new connection: 15 is high; 0 is low.
PREC	8	Send Precedence, Value for send precedence level for the new connection: 15 is high; 0 is low.
TCC	16	TCC, Value for TCC of the new connection with range 0-512.
FORSUB	16	Foreign Subscriber ID.
FORPRT	16	Foreign Port ID, Value for foreign port ID for new connection: bits 1-11 = user ID; bits 12-15 = function suffix; bit 0 = static/dynamic port indicator.
LOCPRT	16	Local Port ID, Value for local port ID for new connection: bits 1-11 = user ID; bits 12-15 = function suffix; bit 0 = static/dynamic port indicator.
LETSIZ	8	Letter Size, Code for typical letter size (in bytes) to be sent on the new connection: 0 = 1-64; 1 = 65-128; 2 = 129-192; 3 = 193-256; 4 = 257-320; 5 = 321-384; 6 = 385-448; 7 = 449 and greater.
SPD	8	Line Speed, Code for input line speed (bps) for host-CCU channel for this user: 0 = 150 or less; 1 = 300; 2 =

MNEMONIC	# OF BITS	DESCRIPTION
		600; 3 = 1200; 4 = 2400; 5 = 4800; 6 = 9600; 7 = 19200 and more.

EVENT SPECIFICATION

EVENT NAME: OPEN RETURN EVENT
MNEMONIC: OOPNRT

CPCI: MCCU

SENDER: TCP

RECEIVER: THP

PURPOSE:

This event informs THP of the local connection name (LCN) for the possible connection, if the Open event was valid, or returns an error status, if the Open event was rejected by TCP.

REASON:

THP must be informed if the open request was invalid so that the user can be informed of an error, or, in the case of a successful open request, the characteristics option can begin. This option is performed only if the user open command or auto-open feature caused the open request to TCP.

COMMENTS:

This event does not mean that the connection has been established. In fact, the three-way handshake will not take place until the first Send event from THP. The Send event queued as a result of the Open Return event (characteristics option record) will be the first Send event and, thus, will cause the three-way handshake.

VERSION: 26-FEB-79

OPEN RETURN EVENT
29-SEP-78

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
1	TRANID								OPCODE								0
3	LCN																2
5	OPNSTA																4
7	INIWND																6

OPEN RETURN EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier. Value for Open Return event = 04.
TRANID	8	Transaction ID. Sequence number of the corresponding Open event with range 1-255.
LCN	16	Local Connection Name. Internal THP-TCP connection identification value with range 1-32.
OPNSTA	16	Status of Open event processing: 0=no error; 2=connection specified in open event exists; 3=unknown user; 6=insufficient resources for new connection; 8=security error; 9=precedence error; 10=TCC error.
INI-WD	16	Initial Send window. Value which provides the initial THP letter ceiling in number of letters which may be outstanding to TCP (Send Return events not received). Possible values are: 0 = Open event was partially specified (listen request), THP Send events are blocked; 1 = Open event was fully specified (open request), THP may send one letter; initial value will not be more than 1.

EVENT SPECIFICATION

EVENT NAME: PREEMPT EVENT
MNEMONIC: CPRMPT

CPCI: MCCU

SENDER: TCP

RECEIVER: TMP

PURPOSE:

This event notifies TMP that a new connection is being established in place of a preempted one, which has been closed. TMP will have received an unsolicited Close Return event with preemption as the close status.

REASON:

The Preempt event acts as an Open Complete event for TMP, i.e., upon receipt of the Preempt event, TMP will send the characteristics option record. TMP reacts the same way to an Open Complete event when the connection is "listening".

COMMENTS:

TMP requires the LCN of the preempted connection for coordination purposes in finishing the close processing for that connection.

VERSION: 08-FEB-79

PREEMPT EVENT
08-FEB-79

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	LCN		2
5	PREC	SEC	4
7	TCC		6
9	FORSUB		8
11	FORPRT		10
13	OLDLCN		12

PREEMPT EVENT
08-FEB-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Preempt event = 47.
TRANID	8	Transaction ID, This byte is always zero for the Preempt event.
LCN	16	Local Connection Name, Internal THP-TCP connection identification with range 1-32, This field contains LCN for new connection.
SEC	8	Connection Security Level, 15 is high; 0 is low.
PREC	8	Connection Precedence Level, 15 is high; 0 is low.
TCC	16	Connection TCC, Binary value with range 0-512.
FORSUB	16	Foreign Subscriber ID.
FORPRT	16	Foreign Port ID.
OLDLCN	16	Old Local Connection Name, LCN of the preempted connection.

EVENT SPECIFICATION

EVENT NAME: RECEIVE EVENT
MNEMONIC: ORCEIV

CPCI: MCCII

SENDER: TCP

RECEIVER: TMP

PURPOSE:

This event passes a letter of data to TMP for output to the user on the specified connection.

REASON:

TCP has received data from the network that must be sent to the user.

COMMENTS:

The triplet buffer, passed via this event, contains buffer IDs for each incoming segment that makes up the TMP letter. Each group of 3 words represents one segment with the first word in the triplet buffer being the number of segments in the letter and the second word in the triplet buffer being reserved for future use.

VERSION: 29-SEP-78

RECEIVE EVENT
29-SEP-78

```
BYTE  15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0  BYTE
    +-------------------------------+-------------------------------+
    !                               !                               !
  1 !            TRANID             !            OPCODE             !   0
    !                               !                               !
    +-------------------------------+-------------------------------+
    !                                                               !
  3 !                              LCN                              !   2
    !                                                               !
    +---------------------------------------------------------------+
    !                                                               !
  5 !                             BUFID                             !   4
    !                                                               !
    +---------------------------------------------------------------+
```

RECEIVE EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Receive event = 42.
TRANID	8	Transaction ID, Sequence number for this event, for coordination of return event, with range 1-255.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.
BUFID	16	Buffer ID, Tag used to map to the triplet buffer containing information about segments to be processed: word 0 = number of segments; word 1 = reserved for future use; words 2-n, in groups of 3-word entries: first word = segment buffer ID; second word = offset to first data byte; third word = number of bytes of data to be processed in segment.

EVENT SPECIFICATION

EVENT NAME: RECEIVE RETURN EVENT
MNEMONIC: ORCVRT

CPCI: MCCU

SENDER: THP

RECEIVER: TCP

PURPOSE:

This event notifies TCP that a letter has been processed for delivery to the user on the connection referenced in the corresponding Receive event.

REASON:

THP has dequeued a Receive event and has processed and moved the data in the THP letter to a "to-user" buffer. The to-user buffer may or may not have been sent to HSI (To User event) for actual output to the user. The triplet and all segment buffers have been processed and released, however.

COMMENTS:

The Receive Return event is used by TCP to determine when an acknowledgement may be sent to remote TCP for data received and delivered to THP.

VERSION: 26-FEB-79

RECEIVE RETURN EVENT
08-FEB-79

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
1	TRANID								OPCODE								0
3	LCN																2

RECEIVE RETURN EVENT
08-FEB-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Receive Return event = 66.
TRANID	8	Transaction ID, Sequence number of the corresponding Receive event with range 1-255.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.

EVENT SPECIFICATION

EVENT NAME: SEND EVENT
MNEMONIC: OSEND

CPCI: MCCU

SENDER: THP

RECEIVER: TCP

PURPOSE:

This event allows THP to send a letter to the network for the referenced connection.

REASON:

THP builds a Send event when it has received enough data from HSI to satisfy the packet release mechanism for this user.

COMMENTS:

TCP will release the letter buffer when Send event processing is complete.

VERSION: 29-SEP-78

SEND EVENT
29-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	LCN		2
5	BUFID		4
7	BUFLEN		6

SEND EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, value for Send event = 65.
TRANID	8	Transaction ID, Sequence number of this event, for coordination with return event, with range 1-255.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.
BUFID	16	Buffer ID, Tag used to map to the buffer containing the letter to be sent to the network.
BUFLEN	16	Buffer Length, Size, in bytes, of the letter to be sent to the network.

EVENT SPECIFICATION

EVENT NAME: SEND RETURN EVENT
MNEMONIC: OSNDRT

CPCI: MCCU

SENDER: TCP

RECEIVER: THP

PURPOSE:

This event provides THP the status of the corresponding Send event processing.

REASON:

This event is returned if TCP finds an error in the preliminary verification of the Send event, if the data referenced in the event is delivered, or if the data is not delivered. THP must be aware of Send event processing completion as the number of outstanding Send events is limited as part of the flow control mechanism for the MCCU.

COMMENTS:

If an "undelivered" status is returned, the connection will be closing. The particular reason for the close will be returned in a Close Return event.

VERSION: 29-SEP-78

SEND RETURN EVENT
29-SEP-78

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	0
3	LCN		2
5	SNDSTA		4

SEND RETURN EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Send Return event = 41.
TRANID	8	Transaction ID, Sequence number of the corresponding Send event with range 1-255.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.
SNDSTA	16	Status of Send event processing: 0=data delivered; 1=nonexistent connection; 4=connection closing when Send event received; 5=send illegal in listen state; 7=data undelivered (connection will be closed).

EVENT SPECIFICATION

EVENT NAME: SEND WINDOW EVENT
MNEMONIC: OSNWND

CPCI: MCSU

SENDER: TCP

RECEIVER: THP

PURPOSE:

This event notifies THP of the new value to be used as the send window on the user-to-network data path. The value represents the maximum number of Send events that THP may have outstanding to TCP at any one time. That is, the number of Send events that have not been acknowledged by TCP with corresponding Send Return events.

REASON:

The send window is part of the flow control used by TCP to ensure that the virtual connection sustains a steady flow of data in either direction with no appreciable backups in any area. If the send window "closes," THP will eventually stop sending From User Return events to MSI, and MSI will eventually begin holding off the host.

COMMENTS:

VERSION: 29-SEP-78

SEND WINDOW EVENT
29-SEP-78

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
1	• • •								OPCODE								0
3	LCN																2
5	NEW-WND																4

SEND WINDOW EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Send Window event = 46.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.
NEWWND	16	New Send Window, New value for send window, indicating the number of letters which may be outstanding to TCP (Send Return events not received), Value must be greater than or equal to the current send window value,

EVENT SPECIFICATION

EVENT NAME: STATUS EVENT
MNEMONIC: OSTATS

CPCI: MCCU

SENDER: THP

RECEIVER: TCP

PURPOSE:

This event requests that TCP return the current status of the specified connection as relates to the local TCP.

REASON:

The user has entered a THP status command and THP believes there is an active connection that is not being held off in THP itself. THP first requests the TCP status of the connection. If that status is "normal," THP will attempt to communicate with the remote THP via a THP status request record.

COMMENTS:

VERSION: 25-NOV-78

STATUS EVENT
05-NOV-78

BYTE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BYTE
1	TRANID								OPCODE								0
3	LCN																2

STATUS EVENT
05-NOV-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Status event = 68.
TRANID	8	Transaction ID, Sequence number for this event, for coordination of the return event, with range 1-255.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.

EVENT SPECIFICATION

EVENT NAME: STATUS RETURN EVENT
MNEMONIC: OSTSRT

CPCI: MCCU

SENDER: TCP

RECEIVER: THP

PURPOSE:

This event notifies THP of the current status of the specified connection as relates to TCP.

REASON:

This event is sent in response to a Status event.

COMMENTS:

VERSION: 29-SEP-78

STATUS RETURN EVENT
29-SEP-78

BYTE	15–8	7–0	BYTE
1	TRANID	OPCODE	0
3	LCN		2
5	SENFLG	RETSTA	4
7	. . .	RECFLG	6

STATUS RETURN EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for Status Return event = 44.
TRANID	8	Transaction ID, Sequence number of the corresponding Status event with range 1-255.
LCN	16	Local Connection Name, Internal TMP-TCP connection identification value with range 1-32.
RETSTA	8	Return Status, 0 = connection established; 1 = connection does not exist; 4 = connection is closing.
SENFLG	8	Send Flag, value indicating current TCP status of the send path; 0 = open; 1 = blocked.
RECFLG	8	Receive Flag, Value indicating current TCP status of the receive path; 0 = open; 1 = blocked, window closed; 2 = blocked, reassembly queue.

EVENT SPECIFICATION

EVENT NAME: THP TIME OUT EVENT
MNEMONIC: OTIME

CPCI: MCCU

SENDER: THP

RECEIVER: THP

PURPOSE:

This event is queued by THP to itself with a preset delay that is reset when data is sent to or received from the network. If the event is dequeued, it notifies THP that the connection is inactive.

REASON:

The event is built when open processing begins (characteristics option processing). The timer is reset when queuing a Send event and when dequeuing a Receive event. If the event is dequeued, the connection is aborted unless there is an option queue and a flush function has occurred on this connection.

COMMENTS:

This event is built using the delay specified for the user in his UPB. The timer ID is kept in the CMB for future reference (cancelling, resetting, or checking). The timer may be cancelled by the user via the time command. There will always be a timer running during option negotiation, however.

VERSION: 29-SEP-78

THP TIME OUT EVENT
29-SEP-78

```
BYTE 15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0   BYT
    +-------------------------------+-------------------------------+
    |                               |                               |
  1 |            . . .              |            OPCODE             |
    |                               |                               |
    +-------------------------------+-------------------------------+
    |                                                               |
  3 |                              LCN                              |
    |                                                               |
    +---------------------------------------------------------------+
```

THP TIME OUT EVENT
29-SEP-78

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, value for THP Time Out event = 38.
LCN	16	Local Connection Name, Internal THP-TCP connection identification value with range 1-32.

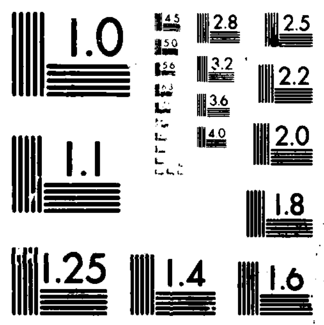
AD-A081 757 WESTERN UNION TELEGRAPH CO MCLEAN VA GOVERNMENT SYST--ETC F/G 17/2
COMPUTER PROGRAM DEVELOPMENT SPECIFICATION TERMINAL-TO-HOST PRO--ETC(U)
FEB 80 DCA200-C-637

UNCLASSIFIED SBIE-AD-E100 340 NL

4 OF 4

AD
A081757

END
DATE
FILMED
4-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

EVENT SPECIFICATION

EVENT NAME: TO USER EVENT
MNEMONIC: OTOUSR

CPCI: MCCU

SENDER: THP

RECEIVER: MSI

PURPOSE:

This event informs MSI that there is a buffer of data to be sent to the user.

REASON:

THP has received network data (via Receive event) and formatted it appropriately for the user in the to-user buffer.

COMMENTS:

A fixed number of bytes will be left blank at the start of the buffer to allow for the insertion of lead protocol characters. Space will also be left at the end of the buffer for trailing protocol characters. This event may also carry control information. One such bit of information has been defined - send break signal (hardware signal).

VERSION: 02-MAR-79

TO USER EVENT
08-FEB-79

BYTE	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	BYTE
1	TRANID	OPCODE	
3	PRILVL	PATHID	
5	BUFTAG		
7	BYTCNT		
9	CONDAT		

TO USER EVENT
28-FEB-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for To User event = 7.
TRANID	8	Transaction ID, Sequence Number of this event, for coordination of return event, with range 1-255.
PATHID	8	Path ID, Internal HSI-THP user identification value with range 1-32.
PRILVL	8	Priority Level, Precedence of data being passed on this connection.
BUFTAG	16	Buffer ID, Tag used to map to the buffer containing data to be sent to the user.
BYTCNT	16	Byte Count, Number of bytes of data to be transmitted.
CONDAT	16	Control Data, Information which cannot be conveyed in the user data stream, such as, hardware break signal, If bit 0 = 1, send hardware break signal on channel.

EVENT SPECIFICATION

EVENT NAME: TO USER RETURN EVENT
MNEMONIC: OTUSRT

CPCI: MCCU

SENDER: HSI

RECEIVER: THP

PURPOSE:

This event notifies THP that the corresponding To User event has been processed and the associated data has been transmitted to the host system.

REASON:

THP must be informed of the To User event processing as the MCCU flow control mechanism allows only a certain number of outstanding To User events.

COMMENTS:

VERSION: 02-MAY-79

TO USER RETURN EVENT
15-FEB-79

BYTE	15–8	7–0	BYTE
1	TRANID	OPCODE	
3	TOUSTA	PATHID	

TO USER RETURN EVENT
15-FEB-79

MNEMONIC	# OF BITS	DESCRIPTION
OPCODE	8	Event Identifier, Value for To User Return event = 36.
TRANID	8	Transaction ID, Sequence number of the corresponding To User event with range 1-255.
PATHID	8	Path ID, Internal HSI-TMP user identification value with range 1-32.
TOUSTA	8	Status of To User event processing: 0 = successful; 1 = unsuccessful.

APPENDIX G

AUTODIN II TRANSMISSION UNITS

There are standard units of transmission for each component of a CCU/TAC, i.e., HSI or TH, THP, TCP, and SIP. It is important to understand the terminology used to describe each, the contents of each, and the relationship of all the units of transmission. The units are discussed below beginning with the "smallest" unit (user text) to the "largest" (SIP or AUTODIN II segment). A visual conception of the relationship is shown in Figure G-1. Figure G-2 shows a "direction of transmission" schematic.

1. user text - This unit of transmission is the smallest for a CCU/TAC. User text is received by HSI or TH on the host-to-CCU or terminal-to-TAC link, respectively. The text may come in a character at a time or in blocks of several characters. HSI/TH accumulates text and passes it to THP via the From User event in the HCCU, and by an equivalent vehicle in the SCCU and TAC. Each event may bring one or many characters, depending on the particular type of user and the link protocol being used. For the purpose of this discussion, and from THP's viewpoint, the unit of transmission called "user text" should be considered as one or more user text characters which, as a unit, have meaning to THP. That is, some characters have special meaning to THP, such as, interrupt function characters, erase line characters, carriage return, linefeed, etc. Other characters have no special meaning to THP, and are accumulated as data to be transmitted to the network in a unit called the THP data record.

Revision 1, July 4, 1979

2. THP record - This unit of transmission is the means by which THPs communicate. That is, each THP record contains information instructing THP as to the required processing. The most common THP record, the data record, consists of user text characters preceded by an 8-bit record mark, an 8-bit record type, and a 16-bit length field. Other THP records consist of an 8-bit record mark, an 8-bit record type, and optional parameter bytes. All THP record types and formats are defined in Appendices A and B of the THP Transportable Specification.

3. THP letter - This unit of transmission is associated with THP and consists of one or more THP records. If only user text characters are being transmitted on the connection, e.g., in binary mode, requiring no other THP control records, there would be one THP data record in the THP letter. If other information is being transmitted, e.g., user text characters followed by an option request, the letter would consist of a THP data record followed by a THP option record. The size of the letter is determined by the various packet release mechanisms of THP (see Reference 9). Letters are passed to TCP via the Send event in the MCCU, and by an equivalent vehicle in an SCCU or TAC.

4. T-segment - This unit of transmission, also known as TCP segment, is associated with TCP and consists of none, one or more, but never partial THP letters, preceded by a T-segment header. The T-segment header is the means by which TCPs communicate, and conveys send/receive flow control information, TCP control information, and addressing information (see Appendix A of the TCP Transportable Specification). The T-segment is passed to SIP for transmission to the SCM via a Send Data event in the MCCU, and the equivalent in an SCCU or TAC.

5. S-segment - This unit of transmission, also known as SIP segment or AUTODIN II segment, is associated with SIP. It is the unit transferred between the CCU or TAC and SCM. The S-segment consists of the T-segment preceded by a binary segment leader (BSL). The BSL contains security, precedence, TCC, addressing, and SIP-SCM control information concerning the S-segment being sent.

PDP-11 MEMORY IMAGE AUTODIN II SEGMENT FORMAT

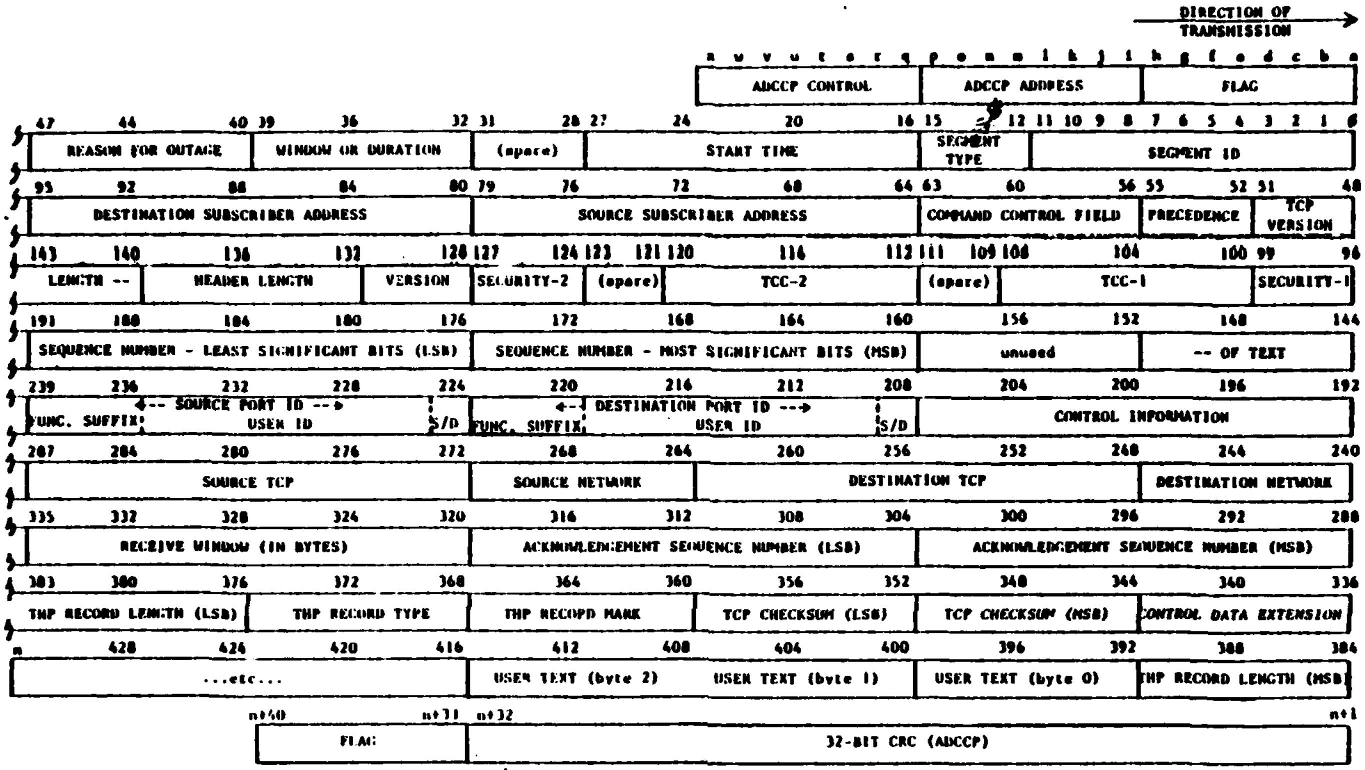
BYTE	.15.14.13.12.11.10. 9 . 8 . 7 . 6 . 5 . 4 . 3 . 2 . 1 . 0 .			BYTE	WORD	
1	SEGMENT TYPE	SEGMENT ID		0	0	BSL
3	(spare)	START TIME		2	1	BSL
5	REASON FOR OUTAGE	WINDOW OR DURATION		4	2	BSL
7	COMMAND CONTROL FIELD	PRECEDENCE	TCP VERSION	6	3	BSL
9	SOURCE SUBSCRIBER ADDRESS			8	4	BSL
11	DESTINATION SUBSCRIBER ADDRESS			10	5	BSL
13	(spare)	TCC-1	SECURITY-1	12	6	BSL
15	SECURITY-2	(spare)	TCC-2	14	7	BSL
17	LENGTH --	HEADER LENGTH	VERSION	16	8	TCP HEADER
19	unused	-- OF TEXT		18	9	TCP HEADER
21	SEQUENCE NUMBER - MOST SIGNIFICANT BITS (MSB)			20	10	TCP HEADER
23	SEQUENCE NUMBER - LEAST SIGNIFICANT BITS (LSB)			22	11	TCP HEADER
25	CONTROL INFORMATION			24	12	TCP HEADER
27	FUNC. SUFFIX	<-- DESTINATION PORT ID --> USER ID	S/D	26	13	TCP HEADER
29	FUNC. SUFFIX	<-- SOURCE PORT ID --> USER ID	S/D	28	14	TCP HEADER
31	DESTINATION TCP (LSB)	DESTINATION NETWORK		30	15	TCP HEADER
33	SOURCE NETWORK	DESTINATION TCP (MSB)		32	16	TCP HEADER
35	SOURCE TCP			34	17	TCP HEADER
37	ACKNOWLEDGEMENT SEQUENCE NUMBER (MSB)			36	18	TCP HEADER
39	ACKNOWLEDGEMENT SEQUENCE NUMBER (LSB)			38	19	TCP HEADER
41	RECEIVE WINDOW (IN BYTES)			40	20	TCP HEADER
43	TCP CHECKSUM (MSB)	CONTROL DATA EXTENSION		42	21	TCP HEADER
		TCP CHECKSUM (LSB)		44	22	TCP HEADER
45	THP RECORD MARK					
47	THP RECORD LENGTH (LSB)	THP RECORD TYPE		46	23	
49	USER TEXT (byte 0)	THP RECORD LENGTH (MSB)		48	24	
51	USER TEXT (byte 2)	USER TEXT (byte 1)		50	25	
⋮	etc.			⋮	⋮	

NOTE: TCP checksum field (bytes 43 and 44) is not used in the CCU/TAC/NCC TCP implementation.

Figure G-1, AUTODIN II Segment Format

Revision 1, July 4, 1979

Figure G-2, AUTODIN II Segment Transmission Schematic



- Binary Segment Leader (BSL) - bits Ø-127
- TCP Header (Note: TAC, CCU, and NCC implementations of TCP do not use checksum field (bits 344 thru 359)) - bits 128-359
- Data (Note: This example illustrates a THP record present in the data stream). - bits 360-n
- ADCCP - 24-bit leader (a thru x)
  - 40-bit trailer (n+1 thru n+40)

Data transmission begins with bit Ø and ends with bit n. (Note: Link control, e.g., ADCCP, is shown outside of bits Ø to n). Within the individual fields, the data is sent from least to most significant bit. For example, if BSL precedence = 8_{10}, the bits would be sent as follows: first bit = Ø, second bit = Ø, third bit = Ø, fourth bit = 1, ..., i.e., precedence = $1ØØØ_2$. Therefore, 1000 ----→ direction of transmission.

Figure G-2. AUTODIN II Segment Transmission Schematic

Revision 1, July 4, 1979

ED

80